

7/5/1 (Item 1 from file: 351)  
DIALOG(R)File 351:Derwent WPI  
(c) 2000 Derwent Info Ltd. All rts. reserv.

011025662 \*\*Image available\*\*

WPI Acc No: 1997-003586/199701

XRPX Acc No: N97-003169

**Program counter debugging method for distributed processing equipment  
furnished with computers - in which debugging information transmitted by  
predetermined computer, is made to receive by same computer**

Patent Assignee: TOSHIBA KK (TOKE )

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 8272644	A	19961018	JP 9577886	A	19950403	199701 B

Priority Applications (No Type Date): JP 9577886 A 19950403

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 8272644	A		29	G06F-011/28	

Abstract (Basic): JP 8272644 A

The method involves generating a request for debugging, using a request generating unit (22). The generated request is then transmitted to the computers by a request transmitting unit (231). When a request is received by the computer, a command input unit (251), inputs a debugging command that is transmitted to the computer by a command transmitting unit (271).

When the command is input, an executing unit (24) executes the debugging command, by which the program counter starts debugging for a number of computers 10i (i=1-n). When debugging is carried out, debugging information is carried back to original computer which issues the debugging command.

ADVANTAGE - Improves transferring efficiency of debugging command and debugging information. Provides computer with high processing capability.

Dwg.2/10

Title Terms: PROGRAM; COUNTER; DEBUG; METHOD; DISTRIBUTE; PROCESS;  
EQUIPMENT; FURNISH; COMPUTER; DEBUG; INFORMATION; TRANSMIT; PREDETERMINED  
; COMPUTER; MADE; RECEIVE; COMPUTER

Derwent Class: T01

International Patent Class (Main): G06F-011/28

International Patent Class (Additional): G06F-009/44; G06F-009/45;

G06F-015/16

File Segment: EPI

7/5/2 (Item 1 from file: 347)  
DIALOG(R)File 347:JAPIO  
(c) 2000 JPO & JAPIO. All rts. reserv.

05317144 \*\*Image available\*\*

DEBUGGING METHOD FOR PROGRAM TO BE EXECUTED BY DISTRIBUTED PROCESSOR

PUB. NO.: 08-272644 JP 8272644 A]

PUBLISHED: October 18, 1996 (19961018)

INVENTOR(s): TAKEWAKI TOSHIKI

APPLICANT(s): TOSHIBA CORP [000307] (A Japanese Company or Corporation), JP  
(Japan)

APPL. NO.: 07-077886 [JP 9577886]

FILED: April 03, 1995 (19950403)

INTL CLASS: [6] G06F-011/28; G06F-009/45; G06F-009/44; G06F-015/16

JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units);  
45.4 (INFORMATION PROCESSING -- Computer Applications)

ABSTRACT

PURPOSE: To perform a debug operation for a program to be distributed and transmitted with no consciousness of distribution.

CONSTITUTION: Each of computers 10-i of a distributed processors includes a request generation part 22 which generates a request with debug to start a program to be debugged, a request sending part 231 which sends the request generated at the part 22 to a computer that includes the debug object program, a debug command input part 251 which issues a debug command to instruct the debug of the program, a command sending part 271 which sends a command to the computer including the object program, a debug execution part 24 which starts the program designated by the received request with debug to set it in a debug enable state and carries out an instructed debug operation to the program designated by the received debug command and a debug information sending part 282 which sends the debug information acquired at the part 24 to the computer that is issued the debug command.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-272644

(43) 公開日 平成8年(1996)10月18日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/28		7313-5B	G 0 6 F 11/28	A
9/45		7737-5B	9/44	5 3 5
9/44	5 3 5		15/16	4 5 0 Z
15/16	4 5 0	7737-5B	9/44	3 2 2 Z 06

審査請求 未請求 請求項の数6 O L (全 29 頁)

(21) 出願番号 特願平7-77886

(22) 出願日 平成7年(1995)4月3日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 武脇 敏晃

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

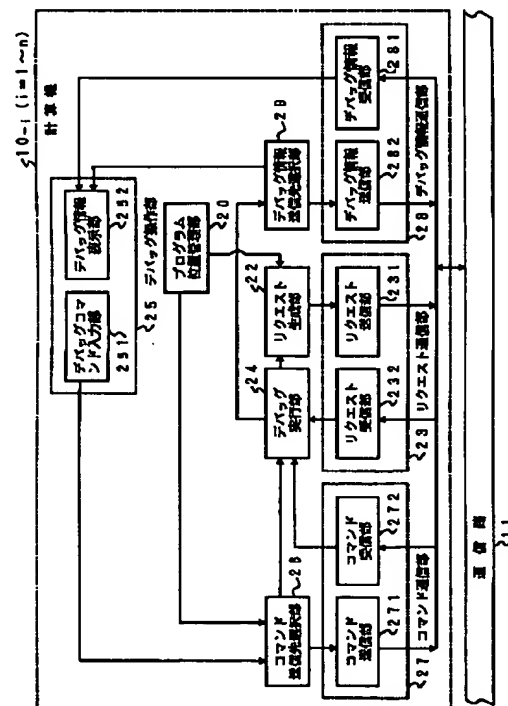
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 分散処理装置で実行されるプログラムのデバッグ方法

(57) 【要約】

【目的】 分散透過なプログラムを分散を意識せずにデバッグ操作可能とする。

【構成】 分散処理装置内の各計算機 10-i に、デバッグ対象プログラムを起動するためのデバッグ付きリクエストを生成するリクエスト生成部 22 と、当該リクエストをデバッグ対象プログラムが存在する計算機に送信するリクエスト送信部 231 と、デバッグ対象プログラムのデバッグを指示するデバッグコマンドを発行するデバッグコマンド入力部 251 と、当該コマンドをデバッグ対象プログラムが存在する計算機に送信するコマンド送信部 271 と、受信したデバッグ付きリクエストの要求するプログラムを起動してデバッグ可能状態とし、受信したデバッグコマンドの指示するプログラムに対して指示されたデバッグ操作を行うデバッグ実行部 24 と、ここで得られたデバッグ情報をデバッグコマンド発行元計算機に送信するデバッグ情報送信部 282 とを備えた構成とする。



## 【特許請求の範囲】

【請求項 1】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、

前記分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報をプログラム位置管理手段により保持しておき、

前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、当該プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

この生成したデバッグ付きリクエストを、前記プログラム位置管理手段の保持情報の示す、当該リクエストにより要求される前記デバッグ対象プログラムが存在する計算機に送信し、

前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態とし、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行して、プログラム位置管理手段の保持情報の示す、当該コマンドにより指示されるデバッグ対象プログラムが存在する計算機に送信し、

前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、このデバッグ操作により得られるデバッグ情報を前記デバッグコマンドの発行元計算機に送信し、この送信されたデバッグ情報が前記発行元計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【請求項 2】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、前記分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報をプログラム位置管理手段により保持しておき、

前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、当該プログラムが存在する計算機を前記プログラム位置管理手段の保持情報をもとに探し、その計算機を送信先として示す送信先情報が付された、当該プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

この生成したデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信し、前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態と

し、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行し、

この発行したデバッグコマンドの指示する前記デバッグ対象プログラムが存在する計算機を前記プログラム位置管理手段の保持情報をもとに探して、その計算機に当該コマンドを送信し、

10 前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、

このデバッグ操作により得られるデバッグ情報を前記デバッグコマンドの発行元計算機に送信し、

この送信されたデバッグ情報が前記発行元計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【請求項 3】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、

20 前記分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報をプログラム位置管理手段により保持すると共に、プログラムのデバッグを指示するデバッグコマンドの送信先計算機を示す情報及びデバッグ情報の送信先計算機を示す情報を送信先管理手段により保持しておき、

前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、当該プログラムが存在する計算機を前記プログラム位置管理手段の保持情報をもとに探し、その計算機を送信先として示す送信先情報及び自身を発行元として示す発行元計算機情報が付された、当該プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

30 この生成したデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信すると共に、当該リクエストの送信先計算機がデバッグコマンドの送信先になるように前記送信先管理手段に保持されているデバッグコマンド送信先情報を変更し、

前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して、当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態とすると共に、当該リクエストに付されている発行元計算機情報の示す計算機がデバッグ情報の送信先となるように前記送信先管理手段に保持されているデバッグ情報送信先情報を変更し、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行し、

50 この発行したデバッグコマンドを、前記送信先管理手段

## 3

に保持されているデバッグコマンド送信先情報の示す計算機に送信し、

前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、このデバッグ操作により得られるデバッグ情報を前記送信先管理手段に保持されているデバッグ情報送信先情報の示す計算機に送信し、

この送信されたデバッグ情報が前記デバッグ情報送信先情報の示す計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【請求項4】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、前記分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報をプログラム位置管理手段により保持すると共に、プログラムのデバッグを指示するデバッグコマンドの送信先計算機を示す情報及びデバッグ情報の送信先計算機を示す情報を送信先管理手段により保持しておき、前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、当該プログラムが存在する計算機を前記プログラム位置管理手段の保持情報をもとに探し、その計算機を送信先として示す送信先情報が付されると共に、前記送信先管理手段に保持されているデバッグ情報送信先情報が付された、当該プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

この生成したデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信すると共に、当該リクエストの送信先計算機がデバッグコマンドの送信先になるように前記送信先管理手段に保持されているデバッグコマンド送信先情報を変更し、前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して、当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態とすると共に、当該リクエストに付されているデバッグ情報送信先情報の示す計算機がデバッグ情報の送信先となるように前記送信先管理手段に保持されているデバッグ情報送信先情報を変更し、

前記デバッグ対象プログラムを起動した計算機において、当該プログラムの実行に伴って他の計算機上の別のデバッグ対象プログラムを起動するために当該他計算機宛てのデバッグ付きリクエストが生成されて当該他計算機に送信された場合には、前記デバッグ対象プログラムの動作している計算機が当該他計算機に変更されたことを通知するための情報を特定のデバッグ情報として生成して、前記送信先管理手段に保持されているデバッグ情報送信先情報の示す計算機に送信し、

## 4

前記デバッグ情報が送信された計算機では、当該デバッグ情報が前記特定のデバッグ情報の場合には、当該デバッグ情報により通知される計算機がデバッグコマンドの送信先になるように前記送信先管理手段に保持されているデバッグコマンド送信先情報を変更し、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って直接または間接に起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行し、

この発行したデバッグコマンドを、前記送信先管理手段に保持されているデバッグコマンド送信先情報の示す計算機に送信し、

前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、このデバッグ操作により得られるデバッグ情報を前記送信先管理手段に保持されているデバッグ情報送信先情報の示す計算機に送信し、

この送信されたデバッグ情報が最初にデバッグ付きリクエストを発行したデバッグ操作を行う計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【請求項5】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、プログラムのインタフェースを定義したインタフェース情報と実行動作を記述した実装コードからなるソースプログラムをプログラム位置管理手段により保持すると共に、プログラムコンパイル時には、前記プログラム位置管理手段に保持されているソースプログラム中のインタフェース情報及び実装コードをもとに、デバッグ対象プログラム起動用のデバッグ付きリクエストを発行するためのリクエスト生成プログラムと、そのリクエストに対応したプログラム起動用のリクエスト起動プログラムを生成して、そのプログラムが存在する計算機を示す存在計算機情報と共に前記プログラム管理手段に保持しておき、

前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、前記プログラム管理手段から前記リクエスト生成プログラムと共に当該デバッグ対象プログラムが存在する計算機の存在計算機情報を探し、そのリクエスト生成プログラム及び存在計算機情報をもとに、その計算機情報が送信先情報として付された、当該デバッグ対象プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

この生成したデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信し、前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して前記プログラム管理手段から

## 5

当該リクエストに対応する前記リクエスト起動プログラムを探し、このリクエスト起動プログラムに従って指定されたデバッグ対象プログラムを起動してデバッグ可能状態とし、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行し、

この発行したデバッグコマンドの指示する前記デバッグ対象プログラムが存在する計算機を前記プログラム管理手段の保持情報をもとに探して、その計算機に当該コマンドを送信し、

前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、

このデバッグ操作により得られるデバッグ情報を前記デバッグコマンドの発行元計算機に送信し、

この送信されたデバッグ情報が前記発行元計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【請求項 6】 複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法であって、

プログラムのインタフェースを定義したインタフェース情報と実行動作を記述した実装コードからなるソースプログラムをプログラム位置管理手段により保持すると共にプログラムのデバッグを指示するデバッグコマンドの送信先計算機を示す情報及びデバッグ情報の送信先計算機を示す情報を送信先管理手段により保持しておき、

プログラムコンパイル時には、前記プログラム位置管理手段に保持されているソースプログラム中のインタフェース情報及び実装コードをもとに、デバッグ対象プログラム起動用のデバッグ付きリクエストを発行するためのリクエスト生成プログラムと、そのリクエストに対応したプログラム起動用のリクエスト起動プログラムを生成して、そのプログラムが存在する計算機を示す存在計算機情報と共に前記プログラム管理手段に保持し、

前記分散処理装置上のある計算機からデバッグ対象プログラムを起動する際には、前記プログラム管理手段から前記リクエスト生成プログラムと共に当該デバッグ対象プログラムが存在する計算機の存在計算機情報を探し、そのリクエスト生成プログラム及び存在計算機情報をもとに、その計算機情報が送信先情報として付されると共に、前記送信先管理手段に保持されているデバッグ情報送信先情報が付された、当該プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成した後、

この生成したデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信すると共に、当該リクエストの送信先計算機がデバッグコマ

## 6

ンドの送信先になるように前記送信先管理手段に保持されているデバッグコマンド送信先情報を変更し、

前記デバッグ付きリクエストが送信された計算機では、当該リクエストを受信して前記プログラム管理手段から当該リクエストに対応する前記リクエスト起動プログラムを探し、このリクエスト起動プログラムに従って指定されたデバッグ対象プログラムを起動してデバッグ可能状態とすると共に、当該リクエストに付されているデバッグ情報送信先情報の示す計算機がデバッグ情報の送信先となるように前記送信先管理手段に保持されているデバッグ情報送信先情報を変更し、

前記デバッグ対象プログラムを起動した計算機において、当該プログラムの実行に伴って他の計算機上の別のデバッグ対象プログラムを起動するために当該他計算機宛てのデバッグ付きリクエストが生成されて当該他計算機に送信された場合には、前記デバッグ対象プログラムの動作している計算機が当該他計算機に変更されたことを通知するための情報を特定のデバッグ情報として生成して、前記送信先管理手段に保持されているデバッグ情報送信先情報の示す計算機に送信し、

前記デバッグ情報が送信された計算機では、当該デバッグ情報が前記特定のデバッグ情報の場合には、当該デバッグ情報により通知される計算機がデバッグコマンドの送信先になるように前記送信先管理手段に保持されているデバッグコマンド送信先情報を変更し、

前記デバッグ付きリクエストを送信した計算機では、当該リクエストに従って直接または間接に起動されているデバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行し、

この発行したデバッグコマンドを、前記送信先管理手段に保持されているデバッグコマンド送信先情報の示す計算機に送信し、

前記デバッグコマンドが送信された計算機では、当該コマンドを受信して、前記デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して当該コマンドにより指示されたデバッグ操作を行い、

このデバッグ操作により得られるデバッグ情報を前記送信先管理手段に保持されているデバッグ情報送信先情報の示す計算機に送信し、

この送信されたデバッグ情報が最初にデバッグ付きリクエストを発行したデバッグ操作を行う計算機で受信されるようにしたことを特徴とする分散処理装置で実行されるプログラムのデバッグ方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数の計算機を備えた分散処理装置で実行されるプログラムのデバッグ方法に関する。

【0002】

【従来の技術】従来から、より処理能力の高い計算機システムを実現するために、分散した複数の計算機（備えた分散処理装置）によって並列に処理を行う分散／並列処理技術の開発が行われている。また近年は、オブジェクト指向の技術を採用した分散オブジェクト指向システムへの関心も高まっており、オブジェクト間通信インタフェースの仕様を標準化する団体もある。

【0003】オブジェクト指向技術の標準化団体であるOMG（Object Management Group）の制定したオブジェクト間通信インタフェースCORBAの仕様に従えば、プログラマからは他計算機上にあるオブジェクトも同一計算機上にあるように見える。このような特徴を分散透過であると呼んでいる。ここでのオブジェクトは、ある機能を持ったプログラムのことであり、以後の説明では、オブジェクトをプログラムと呼ぶこともある。また、以下の説明では、アプリケーションプログラムのように一定の目的を果たすためのプログラムの集合をシステムと表現し、分散透過なプログラム（システム）を分散システムと表現している。

【0004】ここで、分散システム（分散透過なプログラム）に不具合があった場合のデバッグ方法について述べる。説明の仮定として、プログラマがシステム（アプリケーションプログラム）を起動した計算機以外で実行される（オブジェクト指向での）オブジェクト（プログラム）がデバッグ対象であるとする。

【0005】従来のデバッグ方法では、デバッグ対象のオブジェクトが実行されている計算機上にデバッグを起動し、当該デバッグにデバッグ対象オブジェクトを結合することで、オブジェクトの制御を行っている。

【0006】この場合、プログラマは、物理的に分散されている複数の計算機上でデバッグの操作を行うことで、デバッグ作業を実施する。これは、リモートデバッグであっても同じであり、物理的に実行されている計算機をプログラマが認識する必要がある。

【0007】しかしながら、分散透過に見えているプログラムを、本来知る必要がない物理的なオブジェクト位置を使ってデバッグすることは、プログラマからのシステムの見え方が複数存在することになる。これは、デバッグを難しいものにする原因となる。また、本来持っていたオブジェクト（プログラム）がどこにあるかを知る必要がないという特徴をなくすことになる。

【0008】

【発明が解決しようとする課題】上記のように従来のデバッグ方法では、分散透過なプログラムのデバッグであっても、分散処理装置内で物理的に異なる計算機上でデバッグ対象プログラム（オブジェクト）が起動している場合には、それぞれのプログラムが起動している計算機上に対応するデバッグを明示的に起動しなければならないという問題があった。

【0009】また、リモートデバッグのように、他の計

算機からデバッグ情報を取り出し、別計算機上でデバッグ作業を行う場合でも、別の計算機にプログラム（オブジェクト）があることを意識しなければならない、システムの見え方が複数存在するという問題があった。

【0010】本発明は上記事情を考慮してなされたものでその目的は、分散処理装置上における分散透過な単一のシステム（プログラム）のデバッグに必要なデバッグ操作は、プログラマが当該システムを起動した計算機だけで行えるようにし、デバッグ対象プログラムが分散処理装置上で分散しているか否かをプログラマは意識することなくデバッグ操作が行える分散処理装置で実行されるプログラム（分散透過なプログラム、分散システム）のデバッグ方法を提供することにある。

【0011】

【課題を解決するための手段及び作用】本発明の第1の観点に係る構成は、分散処理装置内の各計算機に、分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報を保持しておくプログラム位置管理手段と、デバッグ対象プログラムを起動するに際して、当該プログラムが存在する計算機をプログラム位置管理手段の保持情報をもとに探し、その計算機を送信先として示す送信先情報が付された、当該プログラムの起動を要求するデバッグ付きリクエストを生成するリクエスト生成手段と、上記生成されたデバッグ付きリクエストを当該リクエストに付されている送信先情報の示す計算機に送信するリクエスト送信手段と、デバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行するデバッグ操作手段と、上記発行されたデバッグコマンドの指示するデバッグ対象プログラムが存在する計算機をプログラム位置管理手段の保持情報をもとに探し、その計算機を当該コマンドの送信先として決定するコマンド送信先選択手段と、この決定された送信先計算機にデバッグ操作手段により発行されたデバッグコマンドを送信するコマンド送信手段と、分散処理装置内のある計算機のリクエスト送信手段から送られたデバッグ付きリクエストを受けて当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態とし、分散処理装置内のある計算機のコマンド送信手段から送られたデバッグコマンドを受けて上記起動されているデバッグ対象プログラムに対して指示されたデバッグ操作を行うデバッグ実行手段と、このデバッグ実行手段でのデバッグ操作に際し、対応するデバッグコマンドの発行元計算機を示す情報を保持しておき、デバッグ実行手段のデバッグ操作によりデバッグ情報が得られた際には、上記保持しておいたデバッグコマンド発行元計算機情報の示す計算機を当該デバッグ情報の送信先として決定するデバッグ情報送信先選択手段と、この決定された計算機に上記得られたデバッグ情報を送信するデバッグ情報送信手段とを備えたことを特徴とするものである。なお、デバッグ操作手段、コ

マンド送信先選択手段及びコマンド送信手段は、必ずしも分散処理装置内の全ての計算機に設けられている必要はなく、少なくとも、ユーザ（プログラマ）がデバッグ操作を行う計算機に設けられていればよい。

【0012】上記第1の観点に係る構成においては、ある計算機（デバッグ操作する計算機）からデバッグ対象プログラムを起動するときに、その計算機内のリクエスト生成手段により、当該プログラムを起動してデバッグ可能状態とするためのデバッグ付きリクエストが生成される。この生成されたデバッグ付きリクエストは、プログラム位置管理手段の保持情報の示す、当該リクエストにより要求されるデバッグ対象プログラムが存在する計算機宛てに送信される。

【0013】さて、デバッグ付きリクエストが送られた計算機のデバッグ実行手段は、当該リクエストの要求するデバッグ対象プログラムを起動する。またデバッグ実行手段は、起動したデバッグ対象プログラムをデバッグ可能状態とするために、そのプログラムを（デバッグプログラムにより実現される）デバッグと結合する。

【0014】一方、デバッグ付きリクエストを送信した計算機では、当該リクエストにより起動されるデバッグ対象プログラムのデバッグを指示するデバッグコマンドがデバッグ操作手段から発行される。このデバッグコマンドは、プログラム位置管理手段の保持情報の示す、当該コマンドにより指示されるデバッグ対象プログラムが存在する計算機宛てに、コマンド送信手段により送信される。

【0015】このデバッグコマンドが送られた計算機のデバッグ実行手段は、先のデバッグ付きリクエストによって起動されているデバッグ対象プログラムに対して当該コマンドに従うデバッグ操作を行う。ここで得られたデバッグ情報はデバッグコマンドの発行元計算機に送信され、当該計算機で受信される。

【0016】このように、上記の構成においては、プログラマがアプリケーションプログラム等のシステムを起動した計算機から、分散処理装置内のある計算機に存在する分散透過なプログラムのデバッグ操作を行うことができ、プログラマはデバッグ対象プログラムが分散しているか否かを何ら意識する必要がない。

【0017】本発明の第2の観点に係る構成は、分散処理装置内の各計算機に、分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報を保持しておくプログラム位置管理手段と、デバッグコマンドの送信先計算機を示す情報及びデバッグ情報の送信先計算機を示す情報であって、初期状態においては自身を示す情報を保持しておく送信先管理手段と、デバッグ対象プログラムを起動するに際して、当該プログラムが存在する計算機をプログラム位置管理手段の保持情報をもとに探し、その計算機を送信先として示す送信先情報が付されると共に、送信先管理手段に保持されているデバッグ

情報送信先情報が付されたデバッグ付きリクエストを生成するリクエスト生成手段と、上記生成されたデバッグ付きリクエストを当該リクエストに付されている前記送信先情報の示す計算機に送信するリクエスト送信手段と、上記生成されたデバッグ付きリクエストの送信先計算機がデバッグコマンドの送信先になるように送信先管理手段のデバッグコマンド送信先情報を変更する第1の変更手段と、分散処理装置内のある計算機のリクエスト送信手段から送信されたデバッグ付きリクエストを受信するリクエスト受信手段と、デバッグ対象プログラムのデバッグ操作のために、当該プログラムのデバッグを指示するデバッグコマンドを発行するデバッグ操作手段と、上記デバッグコマンドの送信先計算機を、前記送信先管理手段に保持されているデバッグコマンド送信先情報をもとに決定するコマンド送信先選択手段と、分散処理装置内のある計算機から送信されたデバッグコマンドを受信するコマンド受信手段と、デバッグ操作手段によりデバッグコマンドが発行された場合には、当該コマンドをコマンド送信先選択手段により決定された計算機に送信するコマンド送信手段と、リクエスト受信手段によりデバッグ付きリクエストが受信された場合には、当該リクエストの要求するデバッグ対象プログラムを起動してデバッグ可能状態とし、この起動したデバッグ対象プログラムの実行に伴って他の計算機上の別のデバッグ対象プログラムを起動するためにリクエスト生成手段によりデバッグ付きリクエストが生成されて当該他計算機に送信された場合には、デバッグ対象プログラムの動作している計算機が当該他計算機に変更されたことを通知するための情報を特定のデバッグ情報として生成し、コマンド受信手段によりデバッグコマンドが受信され、且つ当該受信コマンドの送信先としてコマンド送信先選択手段により自身が決定された場合には、当該受信コマンドを受け取って、デバッグ付きリクエストに従って起動されているデバッグ対象プログラムに対して指示されたデバッグ操作を行うデバッグ実行手段と、リクエスト受信手段によりデバッグ付きリクエストが受信された場合には、当該リクエストに付されているデバッグ情報送信先情報の示す計算機がデバッグ情報の送信先となるように、送信先管理手段のデバッグ情報送信先情報を変更する第2の変更手段と、デバッグ情報の送信先計算機を、送信先管理手段に保持されているデバッグ情報送信先情報をもとに決定するデバッグ情報送信先選択手段と、分散処理装置内のある計算機から送信されたデバッグ情報を受信するデバッグ情報受信手段と、デバッグ実行手段でのデバッグ操作によりデバッグ情報が得られた場合と、デバッグ実行手段により上記特定のデバッグ情報が生成された場合には、当該デバッグ情報をデバッグ情報送信先選択手段により決定された計算機に送信するデバッグ情報送信手段とを備え、分散処理装置内のある計算機のデバッグ情報送信手段から上記特定のデバッグ情報



が送信されて、当該デバッグ情報がデバッグ情報受信手段により受信された場合には、当該デバッグ情報により通知される計算機がデバッグコマンドの送信先になるように、送信先管理手段に保持されているデバッグコマンド送信先情報が第1の変更手段により変更されるように構成されていることを特徴とする。なお、デバッグ操作手段、コマンド送信先選択手段、コマンド送信手段及びデバッグ情報受信手段は、必ずしも分散処理装置内の全ての計算機に設けられている必要はなく、少なくとも、ユーザ（プログラマ）がデバッグ操作を行う計算機に設けられていればよい。

【0018】上記第2の観点に係る構成においては、第1の計算機からの第1のデバッグ付きリクエストにより第2の計算機上の第1のデバッグ対象プログラムが起動され、その第1のデバッグ対象プログラムの実行に伴い、第3の計算機に存在する第2のデバッグ対象プログラムを起動する際には、第2の計算機のリクエスト生成手段により、第3の計算機に存在する第2のデバッグ対象プログラムを起動するための第2のデバッグ付きリクエストが発行される。

【0019】この第2のデバッグ付きリクエストには、第2の計算機を送信先として示す送信先情報と第2の計算機を送信先管理手段に保持されているデバッグ情報送信先情報が付加される。このとき送信先管理手段に保持されているデバッグ情報送信先情報は、第1の計算機から第2の計算機に送られた第1のデバッグ付きリクエストに従って第1のデバッグ対象プログラムが起動された場合に、そのリクエストに付されていた第1の計算機を示すデバッグ情報送信先情報（最初のデバッグ付きリクエストの発行元であるデバッグ操作を行う第1の計算機を示す情報）に変更されている。したがって、上記第2のデバッグ付きリクエストに付されるデバッグ情報送信先情報は第1の計算機を示すことになる。

【0020】さて、第2の計算機から第3の計算機に第2のデバッグ付きリクエストが発行されると、第2の計算機のデバッグ実行手段は、デバッグ対象プログラムの動作している計算機が第3の計算機に変更になったことを通知するための特定デバッグ情報を生成する。このデバッグ情報は、第2の計算機を送信先管理手段に保持されているデバッグ情報送信先情報の示す第1の計算機に送られる。

【0021】第1の計算機では、この特定デバッグ情報が受信されると、第1の計算機を送信先管理手段に保持されているデバッグコマンド送信先情報が、当該デバッグ情報で示される第3の計算機を示すように変更される。

【0022】一方、第2の計算機から第2のデバッグ付きリクエストが送られた第3の計算機では、第2のデバッグ対象プログラムが起動されると共に、第3の計算機を送信先管理手段に保持されているデバッグ情報送信先

情報が、当該リクエストに付されているデバッグ情報送信先情報、即ち最初のデバッグ付きリクエストの発行元であるデバッグ操作を行う第1の計算機を示す情報に変更される。

【0023】以上により、第1の計算機からのデバッグコマンドは、第1の計算機を送信先管理手段に保持されているデバッグコマンド送信先情報の示す第3の計算機に直接送信される。また、このデバッグコマンドに従って第3の計算機で起動されている第2のデバッグ対象プログラムに対するデバッグ操作が行われ、その結果得られるデバッグ情報は、第3の計算機を送信先管理手段に保持されているデバッグ情報送信先情報の示す第1の計算機（デバッグ操作を行う計算機）に直接送信される。

【0024】このように、ある計算機からのリクエストで起動されたプログラムの実行により、他の計算機上の別のプログラムを起動するリクエスト（深いリクエストと呼ぶ）を発行する場合に、そのリクエスト発行元計算機を示す情報ではなくて、送信先管理手段に保持されているデバッグ情報送信先情報をそのままデバッグ情報の送信先情報として用いると共に、この種のリクエストを発行した際に、デバッグ対象プログラムが動作している計算機が変更となったことと、その変更先の計算機とを通知する特定のデバッグ情報をデバッグ操作を行う計算機に送信することで、デバッグコマンドとデバッグ情報の転送効率を向上することができる。

【0025】本発明の第3の観点に係る構成は、前記第1の観点に係る構成におけるプログラム位置管理手段及びリクエスト生成手段に代えて、プログラムのインタフェースを定義したインタフェース情報と実行動作を記述した実装コードからなるソースプログラム、当該プログラムをコンパイルして得られるオブジェクトプログラム、及び前記分散処理装置で実行されるプログラムが存在する計算機を示す存在計算機情報を保持しておくプログラム管理手段、及びソースプログラムのコンパイル時に、当該プログラム中のインタフェース情報及び実装コードをもとに、デバッグ対象プログラム起動用のデバッグ付きリクエストを発行するためのリクエスト生成プログラムと、そのリクエストに対応したプログラム起動用のリクエスト起動プログラムを生成して上記プログラム管理手段に保持するプログラムコンパイル手段を設ける他、デバッグ対象プログラムを制御してデバッグ操作を行うデバッグ実行手段に、デバッグ対象プログラムを起動する際には、上記プログラム管理手段からリクエスト生成プログラムと共に当該デバッグ対象プログラムが存在する計算機が存在計算機情報を探し、そのリクエスト生成プログラム及び存在計算機情報をもとに、その計算機情報が送信先情報として付された、当該デバッグ対象プログラムの起動を要求するリクエストであるデバッグ付きリクエストを生成する機能と、リクエスト受信手段によりデバッグ付きリクエストが受信された場合には、

10

20

30

40

50

プログラム管理手段から当該リクエストに対応するリクエスト起動プログラムを探し、このリクエスト起動プログラムに従って、指定されたデバッグ対象プログラムを起動する機能を持たせたことを特徴とする。

【0026】また、本発明の第4の観点に係る構成は、前記第2の観点に係る構成におけるプログラム位置管理手段及びリクエスト生成手段に代えて、上記第3の観点に係る構成と同様のプログラム管理手段、及びプログラムコンパイル手段を設ける他、デバッグ対象プログラムを制御してデバッグ操作を行うデバッグ実行手段に、デバッグ対象プログラムを起動する際には、上記プログラム管理手段からリクエスト生成プログラムと共に当該デバッグ対象プログラムが存在する計算機の存在計算機情報を探し、そのリクエスト生成プログラム及び存在計算機情報をもとに、その計算機情報が送信先情報として付されると共に、送信先管理手段に保持されているデバッグ情報送信先情報が付されたデバッグ付きリクエストを生成する機能と、リクエスト受信手段によりデバッグ付きリクエストが受信された場合には、プログラム管理手段から当該リクエストに対応するリクエスト起動プログラムを探し、このリクエスト起動プログラムに従って、指定されたデバッグ対象プログラムを起動する機能を持たせたことを特徴とする。

【0027】上記第4及び第5の観点に係る構成においては、プログラムコンパイル手段によるソースプログラムのコンパイル時に、当該プログラム中のインタフェース情報及び実装コードをもとにデバッグ対象プログラム起動用のデバッグ付きリクエストを発行するためのリクエスト生成プログラムと、そのリクエストに対応したプログラム起動用のリクエスト起動プログラムを予め静的に生成してプログラム管理手段に保持しておき、デバッグ対象プログラムを起動する際には、プログラム管理手段に保持されている専用のリクエスト生成プログラムを利用することでデバッグ付きリクエストの発行が速やかに行え、デバッグ付きリクエストを受信した際には、プログラム管理手段に保持されている専用のリクエスト起動プログラムを利用することでデバッグ対象プログラムを起動してデバッグ操作可能状態とする制御が速やかに行えるようになる。なお、プログラム実行時に動的にリクエストを組み立てる汎用的な方式では、対象プログラムが起動できるか否かをチェックしてからリクエストが生成されることから、時間がかかる。また、リクエスト生成のために、実行プログラム内でデータタイプ等が調べられることから、この点でも時間がかかる。

【0028】

【実施例】

【第1の実施例】まず、本発明の第1の実施例につき図面を参照して説明する。

【0029】図1は本発明の第1の実施例に係る分散処理装置のシステム構成を示すブロック図、図2は図1中

の各計算機の機能構成を示すブロック図である。

【0030】図1に示す分散処理装置は、複数の計算機10-1, 10-2, …, 10-nと、これら計算機10-1～10-n間の通信に用いられる通信路11により構成される。

【0031】各計算機10-i (i=1～n)は、図2に示すように、プログラム位置管理部20、リクエスト生成部22、リクエスト通信部23、デバッグ実行部24、デバッグ操作部25、コマンド送信先選択部26、コマンド通信部27、デバッグ情報通信部28及びデバッグ情報送信先選択部29を有している。

【0032】プログラム位置管理部20は、デバッグ対象プログラムの存在場所（具体的には、プログラムのプログラム名と当該プログラムが存在する計算機を示す識別子との組）からなる、各プログラム毎の存在計算機情報を保持するものであり、図示しない主記憶または二次記憶上に確保される。

【0033】リクエスト生成部22は、デバッグ対象プログラムを起動してデバッグ可能状態とするための要求（デバッグ付きリクエスト）を生成するものである。

【0034】リクエスト通信部23は、リクエスト生成部22が生成したリクエストを対象計算機に送信するリクエスト送信部231と、図1の分散処理装置内のある計算機から送られたリクエストを受信するリクエスト受信部232からなる。

【0035】デバッグ実行部24は、デバッグコマンドの指示するデバッグ操作を（デバッグプログラムにより実現される）デバガ（図示せず）により実行するものであり、起動リクエストからデバッグ対象プログラムを起動する操作を含んでいる。

【0036】デバッグ操作部25は、ユーザ（プログラマ）によるデバッグ作業のためのインタフェースをなすもので、デバッグ対象プログラムに対するコマンド（デバッグコマンド）の発行を行うデバッグコマンド入力部251と、デバッグ情報を表示するデバッグ情報表示部252からなる。

【0037】コマンド送信先選択部26は、デバッグ対象プログラムの動作する計算機をデバッグコマンドの送信先として選択するものである。

【0038】コマンド通信部27は、デバッグ対象プログラムが起動されている計算機に対してデバッグコマンドを送信するコマンド送信部271と、デバッグ要求元計算機からのデバッグコマンドを受信するコマンド受信部272からなる。

【0039】デバッグ情報通信部28は、デバッグ要求元計算機からのデバッグ情報を受信するデバッグ情報受信部281と、（デバッグ要求元計算機からのデバッグコマンドに従う）デバッグ実行部24でのデバッグ操作により得られるデバッグ情報をデバッグ要求元計算機に送信するデバッグ情報送信部282からなる。

【0040】デバッグ情報送信先選択部29は、デバッグ要求元計算機をデバッグ情報の送信先計算機として選択するものである。

【0041】次に、図2の構成の計算機10-1~10-nを備えた図1の分散処理装置における動作を、ユーザ（プログラマ）のいる計算機10-1からリクエストを発行し、計算機10-2でプログラムP1を動作させてデバッグを行う場合を例に、（1）リクエスト発行時と（2）デバッグ操作時にわけて説明する。

【0042】（1）リクエスト発行時

本実施例におけるリクエストの発行は、プログラムの動作時に他のプログラムを起動する命令（を実行すること）によって行われる。以下の説明では、デバッグ対象プログラムP1を起動するものとする。

【0043】まず、計算機10-1のリクエスト生成部22は、プログラム位置管理部20に保持されているプログラムの存在場所の情報（存在計算機情報）から、デバッグ対象プログラムP1が存在する計算機（の識別子）をリクエスト（デバッグ付きリクエスト）の送信先計算機（の識別子）として探す。プログラムP1が存在する計算機は、複数であっても構わないが、以下の説明では、計算機10-2だけに存在するものとする。

【0044】ここで、リクエスト生成部22において生成しようとしているリクエストの原形（プログラムP1を起動するための起動リクエスト）を図3（a）に示す。この図3（a）中の第1要素“create”は新しいプログラムの起動命令を、第2要素“P1”は起動対象プログラムP1の識別子（プログラム名）を示している。

【0045】計算機10-1のリクエスト生成部22は、図3（a）のリクエストの原形に、プログラム位置管理部20から探したリクエスト送信先計算機10-2（起動すべきデバッグ対象プログラムが存在する計算機10-2）の識別子（計算機#2）と、その原形中のプログラム識別子（P1）で示されるプログラムP1がデバッグ対象プログラムであることを示す情報“debug”を付加し、図3（b）に示すデバッグ付きリクエストを生成する。

【0046】計算機10-1のリクエスト生成部22により生成されたリクエストは、当該リクエストに含まれるリクエスト送信先計算機識別子（計算機#2）をもとに、リクエスト通信部23内のリクエスト送信部231により通信路11を介して当該識別子の計算機10-2に送信される。このリクエストの送信は、（計算機10-2を示す）送信先アドレスと（計算機10-1を示す）送信元アドレスを付加して行われる。

【0047】計算機10-1のリクエスト送信部231により計算機10-2に送信された（計算機10-1からの）デバッグ付きリクエストは、計算機10-2のリクエスト受信部232で受信され、当該計算機10-2のデバッグ

実行部24に渡される。

【0048】計算機10-2のデバッグ実行部24は、リクエスト受信部232から渡された図3（b）に示すようなデバッグ付きリクエストから、図3（a）に示したオリジナルのリクエスト（リクエストの原形）、即ちプログラムP1を起動するためのリクエストを復元して、当該プログラムP1（デバッグ対象プログラムP1）を起動する。このときデバッグ実行部24は、プログラムP1をデバッグするために、そのプログラムP1とデバッガとを結合する。

【0049】（2）デバッグ操作時

次に、計算機10-1から計算機10-2上のプログラムP1（先のデバッグ付きリクエストによって計算機10-2において起動されているプログラムP1）をデバッグする場合の動作を説明する。

【0050】ここでは、計算機10-1のデバッグ操作部25が有するデバッグコマンド入力部251によりデバッグ対象プログラムP1のデバッグ操作を行うものとする。

【0051】まず、計算機10-1のデバッグコマンド入力部251により、デバッグ対象プログラムP1へのブレークポイントの設定を指示する図4に示すような形式のコマンド（デバッグコマンド）が発行されたものとする。図4中の第1要素“breakpoint”はブレークポイントであることを、第2要素“P1-label-i”はデバッグ対象プログラムP1でのブレークポイントとなる論理位置（label-i）を示している。

【0052】計算機10-1のコマンド送信先選択部26は、デバッグコマンド入力部251により発行された図4に示す形式のデバッグコマンドの第2要素（の“P1”の部分）からデバッグ対象プログラムP1を判断する。そしてコマンド送信先選択部26は、プログラム位置管理部20に保持されているプログラムの存在場所の情報（存在計算機情報）から、自身が判断したデバッグ対象プログラムP1が存在する計算機10-2を探し、その計算機10-2をデバッグコマンドの送信先として決定する。

【0053】さて、上記コマンド送信先選択部26でのコマンド送信先計算機の決定の結果、デバッグコマンド入力部251により発行されたデバッグコマンドは、計算機10-1のコマンド送信部271により通信路11を介して計算機10-2に送信される。このデバッグコマンドの送信は、（計算機10-2を示す）送信先アドレスと（計算機10-1を示す）送信元アドレスを付加して行われる。

【0054】計算機10-1のコマンド送信部271により計算機10-2に送信された（計算機10-1からの）デバッグコマンドは、計算機10-2のコマンド受信部272で受信され、当該計算機10-2のデバッグ実行部24

に渡される。

【0055】 計算機 10-2 のデバッグ実行部 24 は、コマンド受信部 272 から渡された図 4 に示すようなデバッグコマンドの指示に従い、（先に受け取った計算機 10-1 からのデバッグ付きリクエストに従って）起動されているデバッグ対象プログラム P1 のブレークポイントを設定する。

【0056】 また、デバッグ実行部 24 は、設定したブレークポイントに対するデバッグ情報をデバッグ要求元計算機に返信するために、コマンド受信部 272 から渡されたデバッグコマンドの発行元（送信元）計算機 10-1 がデバッグ情報の返信先であることをデバッグ情報送信先選択部 29 に通知して、そのデバッグ情報返信先（デバッグコマンド発行元）計算機情報を内部に保持させる。

【0057】 さて、計算機 10-2 のデバッグ実行部 24 は、デバッグコマンドの指示に従ってデバッグ対象プログラム P1 のブレークポイントを設定したことにより、そのブレークポイントでデバッグ情報を得ると、当該デバッグ情報をデバッグ情報送信先選択部 29 に渡す。

【0058】 計算機 10-2 のデバッグ情報送信先選択部 29 は、デバッグ実行部 24 によって当該選択部 29 内部に保持されたデバッグ情報返信先（デバッグコマンド発行元）計算機情報に従い、デバッグコマンドの発行元計算機 10-1 をデバッグ情報の返信先として選択する。この結果、上記デバッグ情報は、計算機 10-2 のデバッグ情報送信部 282 により通信路 11 を介して計算機 10-1 に送信される。このデバッグ情報の送信は、（計算機 10-1 を示す）送信先アドレスと（計算機 10-2 を示す）送信元アドレスを付加して行われる。

【0059】 計算機 10-2 のデバッグ情報送信部 282 により計算機 10-1 に送信された（計算機 10-2 からの）デバッグ情報は、計算機 10-1 のデバッグ情報受信部 281 で受信される。この受信デバッグ情報は、計算機 10-1 のデバッグ情報表示部 252 により表示される。このデバッグ情報をプリント出力することも可能である。

【0060】 なお、以上の説明では、プログラム位置管理部 20 は各計算機 10-1 ~ 10-n にそれぞれ設けられているものとしたが、システムに 1 つだけ設け、各計算機 10-1 ~ 10-n により共有される構成とすることも可能である。この場合、各計算機 10-1 ~ 10-n はプログラム位置管理部 20 を間接的に持つことになる。

【0061】 また、デバッグ操作部 25、コマンド送信先選択部 26、コマンド送信部 27 及びデバッグ情報受信部 281 も各計算機 10-1 ~ 10-n にそれぞれ設けられているものとしたが、デバッグ操作を行う計算機だけに設けられているものであっても構わない。

【0062】 【第 2 の実施例】 次に、本発明の第 2 の実施例につき図面を参照して説明する。

【0063】 まず、本実施例が前記第 1 の実施例と異なる点は、分散処理装置内の各計算機の機能構成であり、分散処理装置のシステム構成自体は変わらない。そこで、以下の説明では、図 1 の分散処理装置のシステム構成を便宜的に参照し、図 1 中の各計算機 10-i（i = 1 ~ n）の本実施例特有の機能構成を図 5 に示して、前記第 1 の実施例における図 2 の機能構成と異なることを明確にする。

【0064】 本実施例における各計算機 10-i（i = 1 ~ n）は、図 5 に示すように、プログラム位置管理部 30、送信先管理部 31、リクエスト生成部 32、リクエスト通信部 33、デバッグ実行部 34、デバッグ操作部 35、コマンド送信先選択部 36、コマンド通信部 37、デバッグ情報通信部 38 及びデバッグ情報送信先選択部 39 を有している。

【0065】 プログラム位置管理部 30 は、デバッグ対象プログラムの存在場所（存在計算機情報）を保持するものであり、主記憶または二次記憶上に確保される。

【0066】 送信先管理部 31 は、デバッグコマンドの送信先の情報及びデバッグ情報の送信先の情報を保持するものであり、主記憶または二次記憶上に確保される。

【0067】 リクエスト生成部 32 は、デバッグ対象プログラムを起動してデバッグ可能状態とするための要求（デバッグ付きリクエスト）を生成するものである。このリクエストには、後述するように、当該リクエストの送信先計算機の情報（識別子）とデバッグ情報の送信先（返信先）計算機の情報（識別子）が付されている。リクエスト生成部 32 は、生成したリクエストをリクエスト送信部 33 により送信する際に、その送信先計算機 30 の情報をデバッグコマンドの送信先情報として送信先管理部 31 に保持するようになっている。

【0068】 リクエスト通信部 33 は、リクエスト生成部 32 が生成したリクエストを対象計算機に送信するリクエスト送信部 33 1 と、図 1 の分散処理装置内のある計算機から送られたリクエストを受信するリクエスト受信部 33 2 からなる。

【0069】 デバッグ実行部 34 は、デバッグコマンドの指示するデバッグ操作をデバッガにより実行するものであり、起動リクエストからデバッグ対象プログラムを起動する操作を含んでいる。デバッグ実行部 34 は、リクエスト受信部 33 2 からリクエストを受け取った場合に、そのリクエストに付されているデバッグ情報送信先（返信先）情報を送信先管理部 31 に保持するようになっている。

【0070】 デバッグ操作部 35 は、ユーザ（プログラマ）によるデバッグ作業のためのインタフェースをなすもので、デバッグ対象プログラムに対するコマンド（デバッグコマンド）の発行を行うデバッグコマンド入力部 35 1 と、デバッグ情報を表示するデバッグ情報表示部 35 2 からなる。デバッグ情報表示部 35 2 は、他計算

機から送られたデバッグ情報を（デバッグ情報受信部 381 からではなく）デバッグ情報送信先選択部 39 から受け取るようになっている。

【0071】 コマンド送信先選択部 36 は、デバッグコマンドの送信先計算機を選択するものである。このデバッグコマンドの送信先計算機は送信先管理部 31 内のデバッグコマンド送信先情報によって示され、計算機 10-i 自身（自計算機）の場合もある。送信の対象となるデバッグコマンドは、デバッグ操作部 35（内のデバッグコマンド入力部 351）からのデバッグコマンドの場合と、コマンド通信部 37（内のコマンド受信部 372）で受信された他計算機からのデバッグコマンドの場合がある。

【0072】 コマンド通信部 37 は、デバッグ対象プログラムが起動されている計算機に対してデバッグコマンドを送信するコマンド送信部 371 と、デバッグ要求元計算機からのデバッグコマンドを受信するコマンド受信部 372 からなる。コマンド送信部 371 は、受信したデバッグコマンドを（デバッグ実行部 34 ではなくて）コマンド送信先選択部 36 に渡すようになっている。

【0073】 デバッグ情報通信部 38 は、デバッグ要求元計算機からのデバッグ情報を受信するデバッグ情報受信部 381 と、（デバッグ要求元計算機からのデバッグコマンドに従う）デバッグ実行部 34 でのデバッグ操作により得られるデバッグ情報をデバッグ要求元計算機に送信するデバッグ情報送信部 382 からなる。デバッグ情報受信部 381 は、受信したデバッグ情報を（デバッグ情報表示部 352 ではなくて）デバッグ情報送信先選択部 39 に渡すようになっている。

【0074】 デバッグ情報送信先選択部 39 は、デバッグ情報の送信先計算機を選択するものである。このデバッグ情報の送信先計算機は送信先管理部 31 内のデバッグ情報送信先情報によって示され、計算機 10-i 自身（自計算機）の場合もある。

【0075】 次に、図 5 の構成の計算機 10-1 ~ 10-n を備えた図 1 の分散処理装置における動作を、ユーザ（プログラマ）のいる計算機 10-1 からリクエストを発行し、計算機 10-2 でプログラム P1 を動作させてデバッグを行う場合を例に、（1）リクエスト発行時と（2）デバッグ操作時に分けて説明する他、（3）深いリクエスト（あるリクエストにより起動されたプログラムから他の計算機上の別のプログラムを起動するためのリクエスト）が発行される場合についても説明する。

【0076】 （1）リクエスト発行時  
今、リクエスト（デバッグ付きリクエスト）発行前において、計算機 10-1、10-2 の送信先管理部 31 には、デバッグコマンドの送信先及びデバッグ情報の送信先として、いずれも当該計算機 10-1、10-2 自身（自計算機）を示す送信先情報が保持されているものとする。

【0077】 本実施例におけるリクエストの発行は、プ

ログラムの動作時に他のプログラムを起動する命令（を実行すること）によって行われる。以下の説明では、デバッグ対象プログラム P1 を起動するものとする。

【0078】 まず、計算機 10-1 のリクエスト生成部 32 は、プログラム位置管理部 30 に保持されている存在計算機情報から、デバッグ対象プログラム P1 が存在する計算機（の識別子）をリクエスト（デバッグ付きリクエスト）の送信先計算機（の識別子）として探す。これにより、（前記第 1 の実施例と同様に）計算機 10-2 がリクエストの送信先計算機として決定されたものとする。

【0079】 次にリクエスト生成部 32 は、新たなプログラム P1 を起動するための図 6（a）に示す（図 3（a）と同様の）リクエストの原形に、プログラム位置管理部 30 から探したリクエスト送信先計算機 10-2 の識別子（計算機 #2）と、その原形中のプログラム識別子（P1）で示されるプログラム P1 がデバッグ対象プログラムであることを示す情報“debug”を付加する他、リクエストの発行元計算機 10-1 の識別子（計算機 #1）をデバッグ情報の送信先（返信先）計算機を示す識別子（デバッグ情報送信先情報）として更に付加し、図 6（b）に示すデバッグ付きリクエストを生成する。

【0080】 この図 6（b）のデバッグ付きリクエストが、図 3（b）のデバッグ付きリクエストと異なる点は、リクエストの発行元計算機（ここでは計算機 10-1）の識別子が、デバッグ情報の返信先（送信先）計算機を示す情報として新たに付加されていることである。

【0081】 さて、計算機 10-1 のリクエスト生成部 32 により生成された図 6（b）のデバッグ付きリクエストは、当該リクエストに含まれるリクエスト送信先計算機識別子（計算機 #2）をもとに、リクエスト通信部 33 内のリクエスト送信部 331 により通信路 11 を介して当該識別子の計算機 10-2 に送信される。

【0082】 このとき計算機 10-1 のリクエスト生成部 32 は、自身の生成したリクエストの送信先計算機 10-2 がデバッグコマンドの送信先となるように、送信先管理部 31 に保持されているデバッグコマンド送信先情報を（現在の計算機 10-1 自身を示す情報から）変更する。

【0083】 計算機 10-1 のリクエスト送信部 331 により計算機 10-2 に送信された（計算機 10-1 からの）デバッグ付きリクエストは、計算機 10-2 のリクエスト受信部 332 で受信され、当該計算機 10-2 のデバッグ実行部 34 に渡される。

【0084】 計算機 10-2 のデバッグ実行部 34 は、リクエスト受信部 332 から渡された図 6（b）に示すようなデバッグ付きリクエストから、図 6（a）に示したオリジナルのリクエスト（リクエストの原形）、即ちプログラム P1 を起動するためのリクエストを復元して、

当該プログラム P 1（デバッグ対象プログラム P 1）を起動する。このときデバッグ実行部 3 4 は、プログラム P 1 をデバッグするために、そのプログラム P 1 とデバッグとを結合する。

【0085】 計算機 1 0-2 のデバッグ実行部 3 4 は更に、リクエスト受信部 3 3 2 から渡された図 6（b）に示すリクエストの第 3 要素であるデバッグ情報送信先情報（ここでは計算機 1 0-1 の識別子）により示される計算機、即ちリクエストの発行元の計算機 1 0-1（ここではデバッグ操作部 3 5 によりデバッグ操作を行う計算機 1 0-1）がデバッグ情報の送信先となるように、送信先管理部 3 1 に保持されているデバッグ情報送信先情報を（現在の計算機 1 0-2 自身を示す情報から）変更する。

【0086】（2）デバッグ操作時

次に、計算機 1 0-1 から計算機 1 0-2 上のプログラム P 1（先のデバッグ付きリクエストによって計算機 1 0-2 において起動されているプログラム P 1）をデバッグする場合の動作を説明する。

【0087】 ここでは、計算機 1 0-1 のデバッグ操作部 3 5 が有するデバッグコマンド入力部 3 5 1 によりデバッグ対象プログラム P 1 のデバッグ操作を行うものとする。

【0088】 まず、計算機 1 0-1 のデバッグ操作部 3 5 が有するデバッグコマンド入力部 3 5 1 により、デバッグ対象プログラム P 1 へのブレークポイントの設定を指示する図 7（a）に示すような形式のコマンド（デバッグコマンド）が発行されたものとする。この図 7（a）に示すコマンドは、図 4 に示したデバッグコマンドと同じものである。

【0089】 計算機 1 0-1 のコマンド送信先選択部 3 6 は、デバッグコマンド入力部 3 5 1 によりデバッグコマンドが発行されると、送信先管理部 3 1 に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機 1 0-1 での（プログラム P 1 を起動するための）リクエスト発行時における送信先管理部 3 1 内のデバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機 1 0-2 が決定される。

【0090】 計算機 1 0-1 のコマンド送信先選択部 3 6 でのコマンド送信先計算機の決定の結果、デバッグコマンド入力部 3 5 1 により発行されたデバッグコマンドは、計算機 1 0-1 のコマンド送信部 3 7 1 により通信路 1 1 を介して計算機 1 0-2 に送信される。

【0091】 計算機 1 0-1 のコマンド送信部 3 7 1 により計算機 1 0-2 に送信された（計算機 1 0-1 からの）デバッグコマンドは、計算機 1 0-2 のコマンド受信部 3 7 2 で受信され、当該計算機 1 0-2 のコマンド送信先選択部 3 6 に渡される。

【0092】 計算機 1 0-2 のコマンド送信先選択部 3 6 は、コマンド受信部 3 7 2 からデバッグコマンドを受け

取ると、送信先管理部 3 1 に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明かなように、計算機 1 0-2 自身が決定される。

【0093】 この計算機 1 0-2 のコマンド送信先選択部 3 6 でのコマンド送信先計算機の決定の結果、コマンド受信部 3 7 2 で受信された（計算機 1 0-1 からの）デバッグコマンドは当該コマンド送信先選択部 3 6 により同じ計算機 1 0-2 内のデバッグ実行部 3 4 に渡される。

【0094】 計算機 1 0-2 のデバッグ実行部 3 4 は、コマンド送信先選択部 3 6 から渡された（計算機 1 0-1 からの）デバッグコマンド（図 7（a）参照）の指示に従い、（先に受け取った計算機 1 0-1 からのデバッグ付きリクエストに従って）起動されているデバッグ対象プログラム P 1 のブレークポイントを設定する。

【0095】 計算機 1 0-2 のデバッグ実行部 3 4 は、デバッグコマンドの指示に従ってデバッグ対象プログラム P 1 のブレークポイントを設定したことにより、そのブレークポイントでデバッグ情報を得ると、当該デバッグ情報をデバッグ情報送信先選択部 3 9 に渡す。

【0096】 計算機 1 0-2 のデバッグ情報送信先選択部 3 9 は、デバッグ実行部 3 4 からデバッグ情報を受け取ると、送信先管理部 3 1 に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機 1 0-2 でのリクエスト受信時における（デバッグ実行部 3 4 による）送信先管理部 3 1 内のデバッグ情報送信先情報の更新から明かなように、その際のリクエスト発行元計算機である計算機 1 0-1 が決定される。

【0097】 この計算機 1 0-2 のデバッグ情報送信先選択部 3 9 でのデバッグ情報送信先計算機の決定の結果、デバッグ実行部 3 4 から渡されたデバッグ情報は、計算機 1 0-2 のデバッグ情報送信部 3 8 2 により通信路 1 1 を介して計算機 1 0-1 に送信される。

【0098】 計算機 1 0-2 のデバッグ情報送信部 3 8 2 により計算機 1 0-1 に送信された（計算機 1 0-2 からの）デバッグ情報は、計算機 1 0-1 のデバッグ情報受信部 3 8 1 で受信され、当該計算機 1 0-1 のデバッグ情報送信先選択部 3 9 に渡される。

【0099】 計算機 1 0-1 のデバッグ情報送信先選択部 3 9 は、デバッグ情報受信部 3 8 1 からデバッグ情報を受け取ると、送信先管理部 3 1 に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明かなように、計算機 1 0-1 自身が決定される。

【0100】 この計算機 1 0-1 のデバッグ情報送信先選択部 3 9 でのデバッグ情報送信先計算機の決定の結果、デバッグ情報受信部 3 8 1 から渡されたデバッグ情報



は、計算機10-1のデバッグ情報表示部352に送られ、当該表示部352により表示される。

【0101】(3) 深いリクエストが発行される場合に、あるリクエストにより起動されたプログラムから他の計算機上の別のプログラムを起動するためのリクエスト(深いリクエスト)が発行される場合について説明する。

【0102】例えば、計算機10-1からのリクエストにより起動された計算機10-2上のプログラムP1の実行により、計算機10-n上のプログラムP2が起動されるものとする。また、このプログラムP2が起動される前において、計算機10-nの送信先管理部31には、デバッグコマンドの送信先及びデバッグ情報の送信先として、いずれも当該計算機10-n自身(自計算機)を示す送信先情報が保持されているものとする。

【0103】さて、計算機10-2でのプログラムP1の実行に伴って計算機10-n上のプログラムP2を起動する際には、計算機10-2のリクエスト生成部32により、計算機10-n上のプログラムP2を起動するためのデバッグ付きリクエストが生成され、計算機10-nに発行される。このリクエストは、新たなプログラムP2を起動するための図6(c)に示すリクエストの原形に、図6(d)に示すように、リクエスト送信先計算機10-2の識別子(計算機#n)と、その原形中のプログラム識別子(P2)で示されるプログラムP2がデバッグ対象プログラムであることを示す情報“debug”と、リクエストの発行元計算機10-2の識別子(計算機#2)を付加したものである。

【0104】この計算機10-2から計算機10-nへの(プログラムP2を起動するための)リクエスト発行時の動作は、先に述べた計算機10-1から計算機10-2への(プログラムP1を起動するための)リクエスト発行時の動作と同様である。

【0105】したがって、プログラムP2を起動するための図6(d)に示すリクエストの発行元計算機10-2では、当該リクエストの送信先計算機10-nがデバッグコマンドの送信先となるように、送信先管理部31に保持されているデバッグコマンド送信先情報が(現在の計算機10-2を示す情報から)変更される。

【0106】同様に、計算機10-2からのリクエストの受信側計算機10-nでは、当該リクエストの発行元(送信先)計算機10-2がデバッグ情報の送信先となるように、送信先管理部31に保持されているデバッグ情報送信先情報が(現在の計算機10-nを示す情報から)変更される。

【0107】以下では、このような状況の下でのデバッグ操作時の動作を、プログラムP1を起動した計算機10-1から(更に具体的に述べるならば、計算機10-1のデバッグ操作部35が有するデバッグコマンド入力部351により)、計算機10-n上のデバッグ対象プログラ

ムP2をデバッグ操作する場合を例に説明する。

【0108】まず、計算機10-1のデバッグ操作部35が有するデバッグコマンド入力部351により、デバッグ対象プログラムP2へのブレークポイントの設定を指示する図7(b)に示すようなコマンド(デバッグコマンド)が発行されたものとする。図7(b)中の第1要素“breakpoint”はブレークポイントであることを、第2要素“P2-label-j”はデバッグ対象プログラムP2でのブレークポイントとなる論理位置(label-j)を示している。

【0109】計算機10-1のコマンド送信先選択部36は、デバッグコマンド入力部351により図7(b)のデバッグコマンドが発行されると、送信先管理部31に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機10-1での(プログラムP1を起動するための)リクエストの発行時における送信先管理部31内のデバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機10-2が決定される。

【0110】この場合、図7(b)のデバッグコマンドは、計算機10-1のコマンド送信部371により計算機10-2に送信される。

【0111】計算機10-1のコマンド送信部371により計算機10-2に送信された(計算機10-1からの)デバッグコマンド(図7(b)参照)は、計算機10-2のコマンド受信部372で受信され、当該計算機10-2のコマンド送信先選択部36に渡される。

【0112】計算機10-2のコマンド送信先選択部36は、コマンド受信部372からデバッグコマンドを受け取ると、送信先管理部31に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機10-2での(プログラムP2を起動するための)リクエスト発行時における送信先管理部31内のデバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機10-nが決定される。

【0113】この場合、計算機10-2のコマンド受信部372で受信された計算機10-1からのデバッグコマンド(図7(b)参照)は、同じ計算機10-2のコマンド送信部371により計算機10-nに送信される。

【0114】計算機10-2のコマンド送信部371により計算機10-nに送信された(計算機10-1からの)デバッグコマンド(図7(b)参照)は、計算機10-nのコマンド受信部372で受信され、当該計算機10-nのコマンド送信先選択部36に渡される。

【0115】計算機10-nのコマンド送信先選択部36は、コマンド受信部372からデバッグコマンド(図7(b)参照)を受け取ると、送信先管理部31に保持されているデバッグコマンド送信先情報から、デバッグコ

マンドの送信先計算機を決定する。ここでは、先に述べたプログラムP2を起動するためのリクエストの発行前の前提条件から明らかなように、計算機10-n自身が決定される。

【0116】この場合、計算機10-nのコマンド受信部372で受信されたデバッグコマンド(図7(b)参照)は、コマンド送信先選択部36により同じ計算機10-2内のデバッグ実行部34に渡される。

【0117】計算機10-nのデバッグ実行部34は、コマンド送信先選択部36から渡された(プログラムP2をデバッグするための)図7(b)に示すデバッグコマンドの指示に従い、(先に受け取った計算機10-2からのデバッグ付きリクエストに従って)起動されているデバッグ対象プログラムP2のブレークポイントを設定する。

【0118】計算機10-nのデバッグ実行部34は、図7(b)に示すデバッグコマンドの指示に従ってデバッグ対象プログラムP2のブレークポイントを設定したことにより、そのブレークポイントでデバッグ情報を得ると、当該デバッグ情報をデバッグ情報送信先選択部39に渡す。

【0119】計算機10-nのデバッグ情報送信先選択部39は、デバッグ実行部34からデバッグ情報を受け取ると、送信先管理部31に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-nでの(プログラムP2を起動するための)リクエスト受信時における(デバッグ実行部34による)送信先管理部31内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-2が決定される。

【0120】この場合、計算機10-nのデバッグ実行部34から渡されたデバッグ情報は、同じ計算機10-nのデバッグ情報送信部382により計算機10-2に送信される。

【0121】計算機10-nのデバッグ情報送信部382により計算機10-2に送信されたデバッグ情報は、当該計算機10-2のデバッグ情報受信部381で受信され、同じ計算機10-2のデバッグ情報送信先選択部39に渡される。

【0122】計算機10-2のデバッグ情報送信先選択部39は、デバッグ情報受信部381からデバッグ情報を受け取ると、送信先管理部31に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-2での(プログラムP1を起動するための)リクエストの受信時における(デバッグ実行部34による)送信先管理部31内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-1が決定される。

【0123】この場合、計算機10-2のデバッグ情報受

信部381で受信されたデバッグ情報は、同じ計算機10-2のデバッグ情報送信部382により計算機10-1に送信される。

【0124】計算機10-2のデバッグ情報送信部382により計算機10-1に送信されたデバッグ情報は、計算機10-1のデバッグ情報受信部381で受信され、当該計算機10-1のデバッグ情報送信先選択部39に渡される。

【0125】計算機10-1のデバッグ情報送信先選択部39は、デバッグ情報受信部381からデバッグ情報を受け取ると、送信先管理部31に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明らかなように、計算機10-1自身が決定される。

【0126】この場合、計算機10-1のデバッグ情報受信部381で受信されたデバッグ情報は、デバッグ情報送信先選択部39により同じ計算機10-1のデバッグ情報表示部352に送られ、当該表示部352により表示される。

【0127】以上のように、リクエスト発行がプログラムの制御に従って複数の計算機に渡ってリレー的に行われる場合には、デバッグコマンドとデバッグ情報の転送が各計算機に渡って繰り返されるため、効率的ではない。

【0128】そこで、このデバッグコマンドとデバッグ情報の転送を効率化するための手法について説明する。ここでは、リクエスト発行に違いがあるだけなので、便宜的に図5の構成を参照して、このリクエスト発行に関する部分だけを説明する。

【0129】まず、計算機10-1から発行されたリクエストにより計算機10-2のプログラムP1が起動された後、このプログラムP1の実行に伴って計算機10-2から計算機10-nのプログラムP2を起動する場合、先の例では、計算機10-2のリクエスト生成部32において、図6(d)に示したリクエストが生成されるものとしたが、ここでは、図6(e)に示すようなリクエスト(デバッグ付きリクエスト)が生成されるようにする。

【0130】この図6(e)のリクエストが、図6(d)に示したリクエストと異なる点は、リクエストの第3の要素であるデバッグ情報送信先情報として、リクエストの発行元計算機10-2の識別子(計算機#2)に代えて、当該リクエストの発行時点において(当該リクエストの発行元の)計算機10-2の送信先管理部31に保持されているデバッグ情報送信先計算機の識別子、即ち最初のリクエストの発行元であるデバッグ操作を行う計算機10-1の識別子(計算機#1)が用いられていることである。

【0131】さて、(計算機10-1からのリクエストにより起動された計算機10-2上のプログラムP1から計

10

20

30

40

50



算機 10-n のプログラム P 2 を起動するために) 図 6 (e) に示したリクエストが計算機 10-2 から計算機 10-n に発行されると、計算機 10-2 のデバッグ実行部 34 は、デバッグ対象プログラムの動作している計算機が (計算機 10-2 から) 計算機 10-n に変更になったことを通知するための情報を特定のデバッグ情報として生成し、デバッグ情報送信先選択部 39 に渡す。

【0132】計算機 10-2 のデバッグ情報送信先選択部 39 は、デバッグ実行部 34 から上記特定のデバッグ情報を受け取ると、送信先管理部 31 に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機 10-2 での (プログラム P 1 を起動するための) リクエスト受信時における (デバッグ実行部 34 による) 送信先管理部 31 内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機 10-1 が決定される。

【0133】この場合、計算機 10-2 のデバッグ実行部 34 から渡された特定のデバッグ情報 (デバッグ対象プログラムの動作している計算機の変更を通知するデバッグ情報) は、同じ計算機 10-2 のデバッグ情報送信部 382 により計算機 10-1 に送信される。

【0134】計算機 10-2 のデバッグ情報送信部 382 により計算機 10-1 に送信された上記特定のデバッグ情報は、計算機 10-1 のデバッグ情報受信部 381 で受信され、当該計算機 10-1 のデバッグ情報送信先選択部 39 に渡される。

【0135】計算機 10-1 のデバッグ情報送信先選択部 39 は、デバッグ情報受信部 381 から上記特定のデバッグ情報を受け取ると、当該情報からデバッグ対象プログラムの動作している計算機が計算機 10-n に変更されたことを判断する。この場合、計算機 10-1 のデバッグ情報送信先選択部 39 は、送信先管理部 31 に保持されているデバッグコマンド送信先情報を、上記デバッグ情報で示される計算機 10-n を示すように変更する。

【0136】一方、計算機 10-2 から計算機 10-n に対して発行される図 6 (e) に示したリクエストは、計算機 10-n のリクエスト受信部 332 で受信され、当該計算機 10-n のデバッグ実行部 34 に渡される。

【0137】計算機 10-n のデバッグ実行部 34 は、リクエスト受信部 332 から渡された図 6 (e) のデバッグ付きリクエストから、図 6 (c) のオリジナルのリクエスト (リクエストの原形) を復元して、デバッグ対象プログラム P 2 を起動する。このときデバッグ実行部 34 は、プログラム P 2 をデバッグするために、そのプログラム P 2 とデバッグとを結合する。

【0138】計算機 10-n のデバッグ実行部 34 は更に、リクエスト受信部 332 から渡された図 6 (e) のリクエストの第 3 要素により示される計算機、即ちデバッグ操作部 35 によりデバッグ操作を行う計算機 10-1

がデバッグ情報の送信先となるように、送信先管理部 31 に保持されているデバッグ情報送信先情報を変更する。

【0139】以上の結果、デバッグ操作する計算機 10-1 の (デバッグ操作部 35 が有する) デバッグコマンド入力部 351 からのデバッグコマンドは、計算機 10-n からの特定デバッグ情報をもとに変更された当該計算機 10-1 の送信先管理部 51 内のデバッグコマンド送信先情報に従い、デバッグ対象プログラム P 2 が動作している計算機 10-n に直接送信される。

【0140】また、計算機 10-n で得られたデバッグ情報は、計算機 10-2 から計算機 10-n に対して発行された図 6 (e) のリクエスト (に付加されているデバッグ情報送信先情報) をもとに変更された当該計算機 10-n の送信先管理部 51 内のデバッグ情報送信先情報に従い、直接計算機 10-1 に返されることになる。

【0141】このように、ある計算機からのリクエストで起動されたプログラムの実行により、他の計算機上の別のプログラムを起動するリクエスト (深いリクエスト) を発行する場合に、そのリクエスト発行元計算機の識別子ではなくて、その時点において送信先管理部 31 に保持されているデバッグ情報送信先情報、即ちデバッグ操作を行う計算機 (最初のリクエスト発行元計算機) の識別子をそのままデバッグ情報の送信先情報として用いると共に、この種のリクエストを発行した際に、デバッグ対象プログラムが動作している計算機が変更となったことと、その変更先の計算機とを通知する情報 (特定のデバッグ情報) をデバッグ操作を行う計算機に送信することで、デバッグコマンドとデバッグ情報の転送効率を向上することができる。

【0142】この方式を適用する場合、デバッグコマンドとデバッグ情報の転送が各計算機に渡って繰り返されないで済むため、デバッグ操作を行わない計算機には、デバッグ操作部 55 は勿論、コマンド送信先選択部 36、コマンド送信部 37 及びデバッグ情報受信部 381 を必ずしも設けておく必要はない。

【0143】〔第 3 の実施例〕次に、本発明の第 3 の実施例につき図面を参照して説明する。

【0144】まず、本実施例が前記第 1 の実施例と異なる点は、分散処理装置内の各計算機の機能構成であり、分散処理装置のシステム構成自体は変わらない。そこで、以下の説明では、図 1 の分散処理装置のシステム構成を便宜的に参照し、図 1 中の各計算機 10-i (i = 1 ~ n) の本実施例特有の機能構成を図 8 に示して、前記第 1 の実施例における図 2 の機能構成と異なることを明確にする。

【0145】本実施例における各計算機 10-i (i = 1 ~ n) は、図 8 に示すように、プログラム管理部 40、プログラムコンパイル部 42、リクエスト通信部 43、デバッグ実行部 44、デバッグ操作部 45、コマンド送

信先選択部 4 6、コマンド通信部 4 7、デバッグ情報通信部 4 8 及びデバッグ情報送信先選択部 4 9 を有している。

【0146】プログラム管理部 4 0 は、ソースプログラムのインタフェース情報及び実装コードと、(ソースプログラムをコンパイルして得られる)オブジェクトプログラムと、デバッグ対象プログラムの存在場所(存在計算機情報)を保持するものであり、主記憶または二次記憶上に確保される。

【0147】プログラムコンパイル部 4 2 は、ソースプログラム内のプログラム起動命令に対応する通常実行リクエストとデバッグ付きリクエストを生成し、各リクエストに対応する起動プログラムを生成するものである。

【0148】リクエスト通信部 4 3 は、デバッグ実行部 4 4 で生成されたリクエストを対象計算機に送信するリクエスト送信部 4 3 1 と、他の計算機からのデバッグ対象プログラムを起動するリクエストを受信するリクエスト受信部 4 3 2 からなる。

【0149】デバッグ実行部 4 4 は、デバッグコマンドの指示するデバッグ操作をデバッガにより実行するものであり、起動リクエストからデバッグ対象プログラムを起動する操作を含んでいる。

【0150】デバッグ操作部 4 5 は、ユーザ(プログラマ)によるデバッグ作業のためのインタフェースをなすもので、デバッグ対象プログラムに対するコマンド(デバッグコマンド)の発行を行うデバッグコマンド入力部 4 5 1 と、デバッグ情報を表示するデバッグ情報表示部 4 5 2 からなる。

【0151】コマンド送信先選択部 4 6 は、デバッグ対象プログラムの動作する計算機をデバッグコマンドの送信先として選択するものである。

【0152】コマンド通信部 4 7 は、デバッグ対象プログラムが起動されている計算機に対してデバッグコマンドを送信するコマンド送信部 4 7 1 と、デバッグ要求元計算機からのデバッグコマンドを受信するコマンド受信部 4 7 2 からなる。

【0153】デバッグ情報通信部 4 8 は、デバッグ要求元計算機からのデバッグ情報を受信するデバッグ情報受信部 4 8 1 と、(デバッグ要求元計算機からのデバッグコマンドに従う)デバッグ実行部 4 4 でのデバッグ操作により得られるデバッグ情報をデバッグ要求元計算機に送信するデバッグ情報送信部 4 8 2 からなる。

【0154】デバッグ情報送信先選択部 4 9 は、デバッグ要求元計算機をデバッグ情報の送信先計算機として選択するものである。

【0155】以上に述べた図 8 の構成は、前記第 1 の実施例で述べた図 2 の構成におけるプログラム位置管理部 2 0、リクエスト生成部 2 2 及びデバッグ実行部 2 4 に代えて、プログラム管理部 4 0、プログラムコンパイル部 4 2 及びデバッグ実行部 4 4 を設け、コンパイル時に

静的にリクエストを生成しておくことで、デバッグ操作の高速化を図るようにしたものである。このため、他の構成要素であるリクエスト通信部 4 3 (内のリクエスト送信部 4 3 1、リクエスト受信部 4 3 2)、デバッグ操作部 4 5 (内のデバッグコマンド入力部 4 5 1、デバッグ情報表示部 4 5 2)、コマンド送信先選択部 4 6、コマンド通信部 4 7 (内のコマンド送信部 4 7 1、コマンド受信部 4 7 2)、デバッグ情報通信部 4 8 (内のデバッグ情報受信部 4 8 1、デバッグ情報送信部 4 8 2) 及びデバッグ情報送信先選択部 4 9 については、図 2 中のリクエスト通信部 2 3 (内のリクエスト送信部 2 3 1、リクエスト受信部 2 3 2)、デバッグ操作部 2 5 (内のデバッグコマンド入力部 2 5 1、デバッグ情報表示部 2 5 2)、コマンド送信先選択部 2 6、コマンド通信部 2 7 (内のコマンド送信部 2 7 1、コマンド受信部 2 7 2)、デバッグ情報通信部 2 8 (内のデバッグ情報受信部 2 8 1、デバッグ情報送信部 2 8 2) 及びデバッグ情報送信先選択部 2 9 と同様である。

【0156】次に、図 8 の構成の計算機 1 0-1 ~ 1 0-n を備えた図 1 の分散処理装置における動作を、ユーザ(プログラマ)のいる計算機 1 0-1 からリクエストを発行し、計算機 1 0-2 でプログラム P 1 を動作させてデバッグを行う場合を例に、(1)コンパイル時と(2)リクエスト発行時と(3)デバッグ操作時に分けて説明する。

【0157】(1)コンパイル時

まず、ソースプログラムは、プログラム間の(メッセージ通信の)インタフェースを定義したインタフェース情報と実行動作を記述した実行コードからなる。

【0158】図 9 は、以下の説明で使用するソースプログラム P 1 の一例を示すものである。このプログラム P 1 は、当該プログラム P 1 からプログラム P 2 を起動する命令“fork (P 2 (A 1, ..., A M))”を含んでいる。また、プログラム P 1 には、入力或いは出力のデータのタイプ等を定義するためのプログラム P 1 及び P 2 のインタフェース情報が付されている。この図 9 に示したソースプログラム P 1 をプログラムコンパイル部 4 2 によりコンパイルするときには、当該プログラム P 1 のインタフェース情報をプログラム管理部 4 0 に登録する。

【0159】プログラムコンパイル部 4 2 は、実装コードの中に含まれるプログラム起動命令に対しては、通常のリクエストを発行するためのコードとデバッグ付きリクエストを発行するためのコード(リクエスト生成プログラム)を、実装コード及びインタフェース情報を使用し静的に生成する。更にプログラムコンパイル部 4 2 は、これらのリクエストに対応したプログラムの起動コード(リクエスト起動プログラム)を実装コード及びインタフェース情報を使用し静的に生成する。ここでは、プログラム P 1 からプログラム P 2 を起動するものとす

る。

【0160】プログラムコンパイル部42により生成されたコード（リクエスト生成プログラム及びリクエスト起動プログラム）は、プログラム管理部40に保持される。更に、プログラムが保持されている（存在する）存在計算機情報もプログラム管理部40に保持される。

【0161】ここで、プログラムコンパイル部42によるコンパイルを行う計算機と、コンパイルされたプログラムが存在する（動作する）計算機とは必ずしも一致していなくても構わないが、その場合には、コンパイルされたプログラムを計算機間通信等により目的の計算機のプログラム管理部40に転送して保持しておく必要がある。また、上記存在計算機情報は、各計算機10-1～10-nのプログラム管理部40に保持しておく必要がある。勿論、この存在計算機情報を、各計算機10-1～10-nが共有する構成とすることも可能である。

【0162】（2）リクエスト発行時

本実施例におけるリクエストの発行は、プログラムの動作時に他のプログラムを起動する命令（を実行すること）によって行われる。以下の説明では、計算機10-1からプログラムP1を起動するものとする。また、このプログラムP1は前記第1及び第2実施例と同様に計算機10-2上に存在するものとする。

【0163】まず、計算機10-1のデバッグ実行部44は、起動命令に対応するリクエスト生成コードをプログラム管理部40から探し、これを実行することでリクエストを生成する。このリクエスト生成コード（リクエスト生成プログラム）は、本実施例においては、プログラム管理部40に保持されているプログラムP1の存在計算機情報、即ち計算機10-2を示す識別子（計算機#2）を取り出し、それをリクエスト送信先計算機の識別子（計算機#2）として、図3（a）に示したプログラムの起動を指定する命令“create”と起動対象となるプログラムP1の識別子（プログラム名）からなるリクエストの原形に付加すると共に、その原形中のプログラム識別子（P1）で示されるプログラムP1がデバッグ対象プログラムであることを示す情報“debug”を更に付加して、デバッグ付きリクエストを生成するためのものである。したがって、このリクエスト生成コードをデバッグ実行部44が実行することで、デバッグ対象プログラムP1を起動するための図3（b）に示したデバッグ付きリクエストが生成される。

【0164】計算機10-1のデバッグ実行部44により生成された図3（b）のリクエストは、当該リクエストに含まれるリクエスト送信先計算機識別子（計算機#2）をもとに、リクエスト通信部43内のリクエスト送信部431により当該識別子の計算機10-2に送信される。

【0165】計算機10-1のリクエスト送信部431により計算機10-2に送信されたデバッグ付きリクエスト

は、当該計算機10-2のリクエスト受信部432で受信され、同じ計算機10-2のデバッグ実行部44に渡される。

【0166】計算機10-2のデバッグ実行部44は、リクエスト受信部432から渡された図3（b）のデバッグ付きリクエストをもとに、これに対応するプログラムP1を起動するためのコード（リクエスト起動プログラム）を計算機10-2のプログラム管理部40から探し、これを実行することで、デバッグ対象プログラムP1の起動、実行を行う。このときデバッグ実行部34は、プログラムP1をデバッグするために、そのプログラムP1とデバッガとを結合する。

【0167】（3）デバッグ操作時

次に、計算機10-1から計算機10-2上のプログラムP1（先のデバッグ付きリクエストによって計算機10-2において起動されているプログラムP1）をデバッグする場合の動作を説明する。

【0168】ここでは、計算機10-1のデバッグ操作部45が有するデバッグコマンド入力部451によりデバッグ対象プログラムP1のデバッグ操作を行うものとする。

【0169】まず、計算機10-1のデバッグコマンド入力部451により、デバッグ対象プログラムP1へのブレークポイントの設定を指示する図4に示したデバッグコマンドが発行されたものとする。

【0170】計算機10-1のコマンド送信先選択部46は、デバッグコマンド入力部451により発行された図4に示すデバッグコマンドの第2要素（の“P1”の部分）からデバッグ対象プログラムP1を判断する。そしてコマンド送信先選択部46は、プログラム位置管理部40に保持されている存在計算機情報から、自身が判断したデバッグ対象プログラムP1が存在する計算機10-2を探し、その計算機10-2をデバッグコマンドの送信先として決定する。

【0171】この場合、デバッグコマンド入力部451により発行された図4のデバッグコマンドは、計算機10-1のコマンド送信部471により計算機10-2に送信される。

【0172】計算機10-1のコマンド送信部471により計算機10-2に送信されたデバッグコマンドは、当該計算機10-2のコマンド受信部472で受信され、同じ計算機10-2のデバッグ実行部44に渡される。

【0173】計算機10-2のデバッグ実行部44は、コマンド受信部472から渡された図4のデバッグコマンドの指示に従い、（先に受け取った計算機10-1からのデバッグ付きリクエストに従って）起動されているデバッグ対象プログラムP1のブレークポイントを設定する。

【0174】また、計算機10-2のデバッグ実行部44は、設定したブレークポイントに対するデバッグ情報を

デバッグ要求元計算機に返信するために、コマンド受信部 472 から渡されたデバッグコマンドの発行元（送信元）計算機 10-1 がデバッグ情報の返信先であることをデバッグ情報送信先選択部 49 に通知して、そのデバッグ情報返信先（デバッグコマンド発行元）計算機情報を内部に保持させる。

【0175】さて、計算機 10-2 のデバッグ実行部 44 は、デバッグコマンドの指示に従ってデバッグ対象プログラム P1 のブレークポイントを設定したことにより、そのブレークポイントでデバッグ情報を得ると、当該デ

バッグ情報をデバッグ情報送信先選択部 49 に渡す。  
【0176】計算機 10-2 のデバッグ情報送信先選択部 49 は、デバッグ実行部 44 によって当該選択部 49 内部に保持されたデバッグ情報返信先（デバッグコマンド発行元）計算機情報に従い、デバッグコマンドの発行元計算機 10-1 をデバッグ情報の返信先として選択する。この結果、上記デバッグ情報は、計算機 10-2 のデバッグ情報送信部 482 により計算機 10-1 に送信される。

【0177】計算機 10-2 のデバッグ情報送信部 482 により計算機 10-1 に送信されたデバッグ情報は、当該計算機 10-1 のデバッグ情報受信部 481 で受信される。この受信デバッグ情報は、計算機 10-1 のデバッグ情報表示部 452 により表示される。

【0178】なお、以上の説明では、デバッグ操作部 45、コマンド送信先選択部 46、コマンド送信部 47 及びデバッグ情報受信部 481 が各計算機 10-1 ~ 10-n にそれぞれ設けられているものとしたが、デバッグ操作を行う計算機だけに設けられているものであっても構わない。

【0179】〔第 4 の実施例〕次に、本発明の第 4 の実施例につき図面を参照して説明する。

【0180】まず、本実施例が前記第 1 の実施例と異なる点は、分散処理装置内の各計算機の機能構成であり、分散処理装置のシステム構成自体は変わらない。そこで、以下の説明では、図 1 の分散処理装置のシステム構成を便宜的に参照し、図 1 中の各計算機 10-i（i=1 ~ n）の本実施例特有の機能構成を図 10 に示して、前記第 1 の実施例における図 2 の機能構成と異なることを明確にする。

【0181】本実施例における各計算機 10-i（i=1 ~ n）は、図 10 に示すように、プログラム管理部 50、送信先管理部 51、プログラムコンパイル部 52、リクエスト通信部 53、デバッグ実行部 54、デバッグ操作部 55、コマンド送信先選択部 56、コマンド通信部 57、デバッグ情報通信部 58 及びデバッグ情報送信先選択部 59 を有している。

【0182】プログラム管理部 50 は、ソースプログラムのインタフェース情報及び実装コードと、（ソースプログラムをコンパイルして得られる）オブジェクトプログラムと、デバッグ対象プログラムの存在場所（存在計

算機情報）を保持するものであり、主記憶または二次記憶上に確保される。

【0183】送信先管理部 51 は、デバッグコマンドの送信先の情報及びデバッグ情報の送信先の情報を保持するものであり、主記憶または二次記憶上に確保される。

【0184】プログラムコンパイル部 52 は、ソースプログラム内のプログラム起動命令に対応する通常実行リクエストとデバッグ付きリクエストを生成し、各リクエストに対応する起動プログラムを生成するものである。

【0185】リクエスト通信部 53 は、デバッグ実行部 54 が生成したリクエストを対象計算機に送信するリクエスト送信部 531 と、他の計算機からのデバッグ対象プログラムを起動するリクエストを受信するリクエスト受信部 532 からなる。

【0186】デバッグ実行部 54 は、デバッグコマンドの指示するデバッグ操作をデバッガにより実行するものであり、起動リクエストからデバッグ対象プログラムを起動する操作を含んでいる。デバッグ実行部 54 は、リクエスト受信部 532 からリクエストを受け取った場合に、そのリクエストに付されているデバッグ情報送信先（返信先）情報を送信先管理部 51 に保持するようになっている。

【0187】デバッグ操作部 55 は、ユーザ（プログラマ）によるデバッグ作業のためのインタフェースをなすもので、デバッグ対象プログラムに対するコマンド（デバッグコマンド）の発行を行うデバッグコマンド入力部 551 と、デバッグ情報を表示するデバッグ情報表示部 552 からなる。デバッグ情報表示部 552 は、他計算機から送られたデバッグ情報をデバッグ情報送信先選択部 59 から受け取るようになっている。

【0188】コマンド送信先選択部 56 は、デバッグコマンドの送信先計算機を選択するものである。このデバッグコマンドの送信先計算機は送信先管理部 51 内のデバッグコマンド送信先情報によって示され、計算機 10-i 自身（自計算機）の場合もある。送信の対象となるデバッグコマンドは、デバッグ操作部 55（内のデバッグコマンド入力部 551）からのデバッグコマンドの場合と、コマンド通信部 57（内のコマンド受信部 572）で受信された他計算機からのデバッグコマンドの場合がある。

【0189】コマンド通信部 57 は、デバッグ対象プログラムが起動されている計算機に対してデバッグコマンドを送信するコマンド送信部 571 と、デバッグ要求元計算機からのデバッグコマンドを受信するコマンド受信部 572 からなる。コマンド送信部 571 は、受信したデバッグコマンドをコマンド送信先選択部 56 に渡すようになっている。

【0190】デバッグ情報通信部 58 は、デバッグ要求元計算機からのデバッグ情報を受信するデバッグ情報受信部 581 と、（デバッグ要求元計算機からのデバッグ

コマンドに従う) デバッグ実行部 5 4 でのデバッグ操作により得られるデバッグ情報をデバッグ要求元計算機に送信するデバッグ情報送信部 5 8 2 からなる。デバッグ情報送信部 5 8 2 は、受信したデバッグ情報をデバッグ情報送信先選択部 5 9 に渡すようになっている。

【0191】デバッグ情報送信先選択部 5 9 は、デバッグ情報の送信先計算機を選択するものである。このデバッグ情報の送信先計算機は送信先管理部 5 1 内のデバッグ情報送信先情報によって示される。

【0192】以上に述べた図 10 の構成は、前記第 2 の実施例で述べた図 5 の構成におけるプログラム位置管理部 3 0、リクエスト生成部 3 2 及びデバッグ実行部 3 4 に代えて、プログラム管理部 5 0、プログラムコンパイル部 5 2 及びデバッグ実行部 5 4 を設けることにより、コンパイル時に静的にリクエストを生成しておくことで、前記第 2 の実施例の特徴である(深いリクエストの発行時における)デバッグコマンドとデバッグ情報の転送効率の良さを維持しながら、デバッグ操作の一層の高速化を図るようにしたものである、このため、他の構成要素である送信先管理部 5 1、リクエスト通信部 4 3

(内のリクエスト送信部 5 3 1、リクエスト受信部 5 3 2)、デバッグ操作部 5 5 (内のデバッグコマンド入力部 5 5 1、デバッグ情報表示部 5 5 2)、コマンド送信先選択部 5 6、コマンド通信部 5 7 (内のコマンド送信部 5 7 1、コマンド受信部 5 7 2)、デバッグ情報通信部 5 8 (内のデバッグ情報受信部 5 8 1、デバッグ情報送信部 5 8 2) 及びデバッグ情報送信先選択部 5 9 については、図 5 中の送信先管理部 3 1、リクエスト通信部 4 3 (内のリクエスト送信部 3 3 1、リクエスト受信部 3 3 2)、デバッグ操作部 3 5 (内のデバッグコマンド入力部 3 5 1、デバッグ情報表示部 3 5 2)、コマンド送信先選択部 3 6、コマンド通信部 3 7 (内のコマンド送信部 3 7 1、コマンド受信部 3 7 2)、デバッグ情報通信部 3 8 (内のデバッグ情報受信部 3 8 1、デバッグ情報送信部 3 8 2) 及びデバッグ情報送信先選択部 3 9 と同様である。また、プログラム管理部 5 0 及びプログラムコンパイル部 5 2 は、前記第 3 の実施例で述べた図 8 の構成におけるプログラム管理部 4 0 及びプログラムコンパイル部 4 2 と同様である。

【0193】次に、図 10 の構成の計算機 1 0-1 ~ 1 0-n を備えた図 1 の分散処理装置における動作を、ユーザ(プログラマ)のいる計算機 1 0-1 からリクエストを発行し、計算機 1 0-2 でプログラム P 1 を動作させてデバッグを行う場合を例に、(1) リクエスト発行時と(2) デバッグ操作時に分けて説明する他、(3) 深いリクエストが発行される場合についても説明する。なお、コンパイル時の動作については、前記第 3 の実施例と同様であるため説明を省略する。必要があれば、前記第 3 の実施例における(1) コンパイル時の動作の説明で、プログラム管理部 4 0 及びプログラムコンパイル部

4 2 を、それぞれプログラム管理部 5 0 及びプログラムコンパイル部 5 2 に読み替えられたい。

【0194】(1) リクエスト発行時

今、リクエスト(デバッグ付きリクエスト)発行前において、計算機 1 0-1、1 0-2 の送信先管理部 5 1 には、デバッグコマンドの送信先及びデバッグ情報の送信先として、いずれも当該計算機 1 0-1、1 0-2 自身(自計算機)を示す送信先情報が保持されているものとする。

【0195】本実施例におけるリクエストの発行は、プログラムの動作時に他のプログラムを起動する命令(を実行すること)によって行われる。以下の説明では、デバッグ対象プログラム P 1 を起動するものとする。

【0196】まず、計算機 1 0-1 のデバッグ実行部 5 4 は、起動命令に対応するリクエスト生成コードをプログラム管理部 4 0 から探し、これを実行することで、デバッグ対象プログラム P 1 を起動するための図 6 (b) に示したデバッグ付きリクエストを生成する。このリクエストには、プログラム管理部 4 0 に保持されているプログラム P 1 の存在計算機情報、即ち計算機 1 0-2 を示す識別子(計算機 # 2)がリクエスト送信先計算機の識別子(計算機 # 2)として付加される他、リクエストの発行元計算機 1 0-1 の識別子(計算機 # 1)がデバッグ情報の送信先(返信先)計算機を示す識別子(デバッグ情報送信先情報)として付加されている。

【0197】計算機 1 0-1 のデバッグ実行部 5 4 により生成された図 6 (b) のリクエストは、当該リクエストに含まれるリクエスト送信先計算機識別子(計算機 # 2)をもとに、リクエスト通信部 5 3 内のリクエスト送信部 5 3 1 により当該識別子の計算機 1 0-2 に送信される。

【0198】このとき計算機 1 0-1 のデバッグ実行部 5 4 は、自身の生成したリクエストの送信先計算機 1 0-2 がデバッグコマンドの送信先となるように、送信先管理部 5 1 に保持されているデバッグコマンド送信先情報(現在の計算機 1 0-1 を示す情報から)を変更する。

【0199】計算機 1 0-1 のリクエスト送信部 5 3 1 により計算機 1 0-2 に送信されたデバッグ付きリクエストは、当該計算機 1 0-2 のリクエスト受信部 5 3 2 で受信され、同じ計算機 1 0-2 のデバッグ実行部 5 4 に渡される。

【0200】計算機 1 0-2 のデバッグ実行部 5 4 は、リクエスト受信部 5 3 2 から渡された図 6 (b) のデバッグ付きリクエストをもとに、これに対応するプログラム P 1 を起動するためのコード(リクエスト起動プログラム)を計算機 1 0-2 のプログラム管理部 4 0 から探し、これを実行することで、デバッグ対象プログラム P 1 の起動、実行を行う。このときデバッグ実行部 3 4 は、プログラム P 1 をデバッグするために、そのプログラム P 1 とデバッグとを結合する。

【0201】計算機 1 0-2 のデバッグ実行部 5 4 は更

に、リクエスト受信部532から渡された図6(b)のリクエストの第3要素であるデバッグ情報送信先情報

(ここでは計算機10-1の識別子)により示される計算機、即ちリクエストの発行元の計算機10-1(ここではデバッグ操作部55によりデバッグ操作を行う計算機10-1)がデバッグ情報の送信先となるように、送信先管理部51に保持されているデバッグ情報送信先情報を(現在の計算機10-2自身を示す情報から)変更する。

【0202】(2)デバッグ操作時

次に、計算機10-1から計算機10-2上のプログラムP1(先のデバッグ付きリクエストによって計算機10-2において起動されているプログラムP1)をデバッグする場合の動作を説明する。

【0203】ここでは、計算機10-1のデバッグ操作部55が有するデバッグコマンド入力部551によりデバッグ対象プログラムP1のデバッグ操作を行うものとする。

【0204】まず、計算機10-1のデバッグ操作部55が有するデバッグコマンド入力部551により、デバッグ対象プログラムP1へのブレークポイントの設定を指示する図7(a)に示すようなデバッグコマンドが発行されたものとする。

【0205】計算機10-1のコマンド送信先選択部56は、デバッグコマンド入力部551によりデバッグコマンドが発行されると、送信先管理部51に保持されているデバッグコマンド送信先情報から、当該デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機10-1での(プログラムP1を起動するための)リクエスト発行時における送信先管理部51内のデバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機10-2が決定される。

【0206】この場合、デバッグコマンド入力部551により発行された図7(a)のデバッグコマンドは、計算機10-1のコマンド送信部571により計算機10-2に送信される。

【0207】計算機10-1のコマンド送信部571により計算機10-2に送信されたデバッグコマンドは、当該計算機10-2のコマンド受信部572で受信され、同じ計算機10-2のコマンド送信先選択部56に渡される。

【0208】計算機10-2のコマンド送信先選択部56は、コマンド受信部572から図7(a)のデバッグコマンドを受け取ると、送信先管理部51に保持されているデバッグコマンド送信先情報から、当該デバッグコマンドの送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明らかなように、計算機10-2自身が決定される。

【0209】この場合、コマンド受信部572で受信されたデバッグコマンドは当該コマンド送信先選択部56により同じ計算機10-2内のデバッグ実行部54に渡さ

れる。

【0210】計算機10-2のデバッグ実行部54は、コマンド送信先選択部56から渡されたデバッグコマンド(図7(a)参照)の指示に従い、(先に受け取った計算機10-1からのデバッグ付きリクエストに従って)起動されているデバッグ対象プログラムP1のブレークポイントを設定する。

【0211】計算機10-2のデバッグ実行部54は、デバッグコマンドの指示に従ってデバッグ対象プログラムP1のブレークポイントを設定したことにより、そのブレークポイントでデバッグ情報を得ると、当該デバッグ情報をデバッグ情報送信先選択部59に渡す。

【0212】計算機10-2のデバッグ情報送信先選択部59は、デバッグ実行部54からデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-2でのリクエスト受信時における(デバッグ実行部54による)送信先管理部51内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-1が決定される。

【0213】この場合、デバッグ実行部54から渡されたデバッグ情報は、計算機10-2のデバッグ情報送信部582により計算機10-1に送信される。

【0214】計算機10-2のデバッグ情報送信部582により計算機10-1に送信されたデバッグ情報は、当該計算機10-1のデバッグ情報受信部581で受信され、同じ計算機10-1のデバッグ情報送信先選択部59に渡される。

【0215】計算機10-1のデバッグ情報送信先選択部59は、デバッグ情報受信部581からデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明らかなように、計算機10-1自身が決定される。

【0216】この計算機10-1のデバッグ情報送信先選択部59でのデバッグ情報送信先計算機の決定の結果、デバッグ情報受信部581から渡されたデバッグ情報は、計算機10-1のデバッグ情報表示部552に送られ、当該表示部552により表示される。

【0217】(3)深いリクエストが発行される場合次に、深いリクエストが発行される場合について説明する。

【0218】例えば、計算機10-1からのリクエストにより起動された計算機10-2上のプログラムP1の実行により、計算機10-n上のプログラムP2が起動されるものとする。また、このプログラムP2が起動される前において、計算機10-nの送信先管理部51には、デバッグコマンドの送信先及びデバッグ情報の送信先とし

て、いずれも当該計算機 10-n 自身（自計算機）を示す送信先情報が保持されているものとする。

【0219】さて、計算機 10-2 でのプログラム P 1 の実行に伴って計算機 10-n 上のプログラム P 2 を起動する際には、計算機 10-2 のデバッグ実行部 54 により、計算機 10-n 上のプログラム P 2 を起動するための図 6 (d) に示したデバッグ付きリクエストが生成され、計算機 10-n に発行される。このリクエストには、プログラム管理部 40 に保持されているプログラム P 2 の存在計算機情報、即ち計算機 10-n を示す識別子（計算機 # n）がリクエスト送信先計算機の識別子（計算機 # n）として付加される他、リクエストの発行元計算機 10-2 の識別子（計算機 # 2）がデバッグ情報の送信先（返信先）計算機を示す識別子（デバッグ情報送信先情報）として付加されている。

【0220】この計算機 10-2 から計算機 10-n への（プログラム P 2 を起動するための）リクエスト発行時の動作は、先に述べた計算機 10-1 から計算機 10-2 への（プログラム P 1 を起動するための）リクエスト発行時の動作と同様である。

【0221】したがって、プログラム P 2 を起動するための図 6 (d) に示すリクエストの発行元計算機 10-2 では、当該リクエストの送信先計算機 10-n がデバッグコマンドの送信先となるように、送信先管理部 51 に保持されているデバッグコマンド送信先情報が（現在の計算機 10-2 を示す情報から）変更される。

【0222】同様に、計算機 10-2 からのリクエストの受信側計算機 10-n では、当該リクエストの発行元（送信先）計算機 10-2 がデバッグ情報の送信先となるように、送信先管理部 51 に保持されているデバッグ情報送信先情報が（現在の計算機 10-n を示す情報から）変更される。

【0223】以下では、このような状況の下でのデバッグ操作時の動作を、プログラム P 1 を起動した計算機 10-1 から（更に具体的に述べるならば、計算機 10-1 のデバッグ操作部 55 が有するデバッグコマンド入力部 551 により）、計算機 10-n 上のデバッグ対象プログラム P 2 をデバッグ操作する場合を例に説明する。

【0224】まず、計算機 10-1 のデバッグ操作部 55 が有するデバッグコマンド入力部 551 により、デバッグ対象プログラム P 2 へのブレークポイントの設定を指示する図 7 (b) に示したデバッグコマンドが発行されたものとする。

【0225】計算機 10-1 のコマンド送信先選択部 56 は、デバッグコマンド入力部 551 により図 7 (b) のデバッグコマンドが発行されると、送信先管理部 51 に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機 10-1 での（プログラム P 1 を起動するための）リクエストの発行時における送信先管理部 51 内の

デバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機 10-2 が決定される。

【0226】この場合、図 7 (b) のデバッグコマンドは、計算機 10-1 のコマンド送信部 571 により計算機 10-2 に送信される。

【0227】計算機 10-1 のコマンド送信部 571 により計算機 10-2 に送信されたデバッグコマンド（図 7 (b) 参照）は、当該計算機 10-2 のコマンド受信部 572 で受信され、同じ計算機 10-2 のコマンド送信先選択部 56 に渡される。

【0228】計算機 10-2 のコマンド送信先選択部 56 は、コマンド受信部 572 からデバッグコマンドを受け取ると、送信先管理部 51 に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先の計算機 10-2 での（プログラム P 2 を起動するための）リクエスト発行時における送信先管理部 51 内のデバッグコマンド送信先情報の更新から明らかなように、その際のリクエスト送信先計算機である計算機 10-n が決定される。

【0229】この場合、計算機 10-2 のコマンド受信部 572 で受信された計算機 10-1 からのデバッグコマンド（図 7 (b) 参照）は、同じ計算機 10-2 のコマンド送信部 571 により計算機 10-n に送信される。

【0230】計算機 10-2 のコマンド送信部 571 により計算機 10-n に送信されたデバッグコマンド（図 7 (b) 参照）は、当該計算機 10-n のコマンド受信部 572 で受信され、同じ計算機 10-n のコマンド送信先選択部 56 に渡される。

【0231】計算機 10-n のコマンド送信先選択部 56 は、コマンド受信部 572 からデバッグコマンド（図 7 (b) 参照）を受け取ると、送信先管理部 51 に保持されているデバッグコマンド送信先情報から、デバッグコマンドの送信先計算機を決定する。ここでは、先に述べたプログラム P 2 を起動するためのリクエストの発行前の前提条件から明らかなように、計算機 10-n 自身が決定される。

【0232】この場合、計算機 10-n のコマンド受信部 572 で受信されたデバッグコマンド（図 7 (b) 参照）は、コマンド送信先選択部 56 により同じ計算機 10-2 内のデバッグ実行部 54 に渡される。

【0233】計算機 10-n のデバッグ実行部 54 は、コマンド送信先選択部 56 から渡された（プログラム P 2 をデバッグするための）図 7 (b) のデバッグコマンドの指示に従い、（先に受け取った計算機 10-2 からのデバッグ付きリクエストに従って）起動されているデバッグ対象プログラム P 2 のブレークポイントを設定する。

【0234】計算機 10-n のデバッグ実行部 54 は、図 7 (b) のデバッグコマンドの指示に従ってデバッグ対象プログラム P 2 のブレークポイントを設定したことに



より、そのブレイクポイントでデバッグ情報を得ると、当該デバッグ情報をデバッグ情報送信先選択部59に渡す。

【0235】 計算機10-nのデバッグ情報送信先選択部59は、デバッグ実行部54からデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-nでの（プログラムP2を起動するための）リクエスト受信時における（デバッグ実行部54による）送信先管理部51内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-2が決定される。

【0236】 この場合、計算機10-nのデバッグ実行部54から渡されたデバッグ情報は、同じ計算機10-nのデバッグ情報送信部582により計算機10-2に送信される。

【0237】 計算機10-nのデバッグ情報送信部582により計算機10-2に送信されたデバッグ情報は、計算機10-2のデバッグ情報受信部581で受信され、当該計算機10-2のデバッグ情報送信先選択部59に渡される。

【0238】 計算機10-2のデバッグ情報送信先選択部59は、デバッグ情報受信部581からデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-2での（プログラムP1を起動するための）リクエストの受信時における（デバッグ実行部54による）送信先管理部51内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-1が決定される。

【0239】 この場合、計算機10-2のデバッグ情報受信部581で受信されたデバッグ情報は、同じ計算機10-2のデバッグ情報送信部582により計算機10-1に送信される。

【0240】 計算機10-2のデバッグ情報送信部582により計算機10-1に送信されたデバッグ情報は、計算機10-1のデバッグ情報受信部581で受信され、当該計算機10-1のデバッグ情報送信先選択部59に渡される。

【0241】 計算機10-1のデバッグ情報送信先選択部59は、デバッグ情報受信部581からデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先に述べたリクエスト発行前の前提条件から明らかなように、計算機10-1自身が決定される。

【0242】 この場合、計算機10-1のデバッグ情報受信部581で受信されたデバッグ情報は、デバッグ情報送信先選択部59により計算機10-1のデバッグ情報表

示部552に送られ、当該表示部552により表示される。

【0243】 以上のように、深いリクエストが発行される場合には、デバッグコマンドとデバッグ情報の転送が各計算機に渡って繰り返されるため、効率的ではない。

【0244】 そこで、このデバッグコマンドとデバッグ情報の転送を効率化するための手法について説明する。ここでは、リクエスト発行に違いがあるだけなので、便宜的に図10の構成を参照して、このリクエスト発行に関する部分だけを説明する。

【0245】 まず、計算機10-1から発行されたリクエストにより計算機10-2のプログラムP1が起動された後、このプログラムP1の実行に伴って計算機10-2から計算機10-nのプログラムP2を起動する場合、計算機10-2のデバッグ実行部54は、図6（d）に示したリクエストではなくて、図6（e）に示すようなリクエスト（デバッグ付きリクエスト）を生成する。この図6（e）のリクエストには、第3の要素であるデバッグ情報送信先情報として、リクエストの発行元計算機10-2の識別子（計算機#2）に代えて、当該リクエストの発行時点において計算機10-2の送信先管理部51に保持されているデバッグ情報送信先計算機の識別子、即ち最初のリクエストの発行元であるデバッグ操作を行う計算機10-1の識別子（計算機#1）が付加される。

【0246】 さて、図6（e）のリクエストが計算機10-2から計算機10-nに発行されると、計算機10-2のデバッグ実行部54は、デバッグ対象プログラムの動作している計算機が（計算機10-2から）計算機10-nに変更になったことを通知するための情報を特定のデバッグ情報として生成し、デバッグ情報送信先選択部59に渡す。

【0247】 計算機10-2のデバッグ情報送信先選択部59は、デバッグ実行部54から上記特定のデバッグ情報を受け取ると、送信先管理部51に保持されているデバッグ情報送信先情報から、デバッグ情報の送信先計算機を決定する。ここでは、先の計算機10-2での（プログラムP1を起動するための）リクエスト受信時における（デバッグ実行部54による）送信先管理部51内のデバッグ情報送信先情報の更新から明らかなように、その際のリクエスト発行元計算機である計算機10-1が決定される。

【0248】 この場合、計算機10-2のデバッグ実行部54から渡された特定のデバッグ情報は、計算機10-2のデバッグ情報送信部582により計算機10-1に送信される。

【0249】 計算機10-2のデバッグ情報送信部582により計算機10-1に送信された上記特定のデバッグ情報は、計算機10-1のデバッグ情報受信部581で受信され、当該計算機10-1のデバッグ情報送信先選択部59に渡される。



【0250】 計算機 10-1 のデバッグ情報送信先選択部 59 は、デバッグ情報受信部 581 から上記特定のデバッグ情報を受け取ると、当該情報からデバッグ対象プログラムの動作している計算機が計算機 10-n に変更されたことを判断する。この場合、計算機 10-1 のデバッグ情報送信先選択部 59 は、送信先管理部 51 に保持されているデバッグコマンド送信先情報を、上記デバッグ情報で示される計算機 10-n を示すように変更する。

【0251】 一方、計算機 10-2 から計算機 10-n に対して発行される図 6 (e) に示したリクエストは、計算機 10-n のリクエスト受信部 532 で受信され、当該計算機 10-n のデバッグ実行部 54 に渡される。

【0252】 計算機 10-n のデバッグ実行部 54 は、リクエスト受信部 532 から渡された図 6 (e) のデバッグ付きリクエストをもとに、これに対応するプログラム P2 を起動するためのコード (リクエスト起動プログラム) を計算機 10-n のプログラム管理部 50 から探し、これを実行することで、デバッグ対象プログラム P2 の起動、実行を行う。このときデバッグ実行部 54 は、プログラム P2 をデバッグするために、そのプログラム P2 とデバッグとを結合する。

【0253】 計算機 10-n のデバッグ実行部 54 は更に、リクエスト受信部 532 から渡された図 6 (e) のリクエストの第 3 要素により示される計算機、即ちデバッグ操作部 55 によりデバッグ操作を行う計算機 10-1 がデバッグ情報の送信先となるように、送信先管理部 51 に保持されているデバッグ情報送信先情報を変更する。

【0254】 以上の結果、デバッグ操作する計算機 10-1 の (デバッグ操作部 55 が有する) デバッグコマンド入力部 551 からのデバッグコマンドは、当該計算機 10-1 の送信先管理部 51 に保持されている変更後のデバッグコマンド送信先情報に従い、デバッグ対象プログラム P2 が動作している計算機 10-n に直接送信される。

【0255】 また、計算機 10-n で得られたデバッグ情報は、当該計算機 10-n の送信先管理部 51 に保持されている変更後のデバッグ情報送信先情報に従い、直接計算機 10-1 に返されることになる。

【0256】 この方式を適用する場合、デバッグ操作を行わない計算機には、デバッグ操作部 55 は勿論、コマンド送信先選択部 56、コマンド送信部 57 及びデバッグ情報受信部 581 を必ずしも設けておく必要はない。

【0257】

【発明の効果】 以上詳述したように本発明によれば、デバッグ操作を行う計算機からデバッグ対象プログラムを起動するときに、当該プログラムを起動してデバッグ可能状態とするためのデバッグ付きリクエストが、当該プログラムが実行される計算機宛てに送られることにより、その計算機において当該プログラムが起動されてデバッグ可能状態となり、この状態でデバッグ操作計算機

からデバッグ対象プログラムが起動されている計算機にデバッグコマンドが送られることにより、当該プログラムのデバッグが行われて、そのデバッグ情報がデバッグ操作計算機に送られる構成としたので、分散処理装置上の分散透過なプログラムのデバッグ操作を、実際のプログラムの分散を意識することなく、プログラマがシステムを起動した計算機だけで行うことができる。

【0258】 また、本発明によれば、ある計算機からのデバッグ付きリクエストで起動されたプログラムの実行により、他の計算機上の別のプログラムを起動するデバッグ付きリクエスト (深いリクエスト) を発行する場合に、そのリクエスト発行元計算機を示す情報ではなくて、その計算機 (内の送信先管理手段) に保持されているデバッグ情報送信先情報をそのままデバッグ情報の送信先情報として用いると共に、この種のリクエストを発行した際に、デバッグ対象プログラムが動作している計算機が変更となったことと、その変更先の計算機とを通知する特定のデバッグ情報をデバッグ操作を行う計算機に送信することにより、デバッグコマンドとデバッグ情報の転送効率を向上することができる。

【0259】 また、本発明によれば、ソースプログラムのコンパイル時に、当該プログラム中のインタフェース情報及び実装コードをもとに、デバッグ対象プログラム起動用のデバッグ付きリクエストを発行するためのリクエスト生成プログラムと、そのリクエストに対応したプログラム起動用のリクエスト起動プログラムを静的に生成して、そのプログラムが存在する計算機を示す存在計算機情報と共にプログラム管理手段に保持しておき、デバッグ対象プログラムを起動する際には、プログラム管理手段に保持されている専用のリクエスト生成プログラムを利用することでデバッグ付きリクエストの発行が速やかに行え、デバッグ付きリクエストを受信した際には、プログラム管理手段に保持されている専用のリクエスト起動プログラムを利用することでデバッグ対象プログラムを起動してデバッグ操作可能状態とする制御が速やかに行える。

【図面の簡単な説明】

【図 1】 本発明の第 1 乃至第 4 の実施例で適用される分散処理装置のシステム構成を示すブロック図。

【図 2】 本発明の第 1 の実施例に係る計算機の機能構成を示すブロック図。

【図 3】 同第 1 の実施例及び第 3 の実施例で適用されるリクエストの原形とデバッグ付きリクエストの例を示す図。

【図 4】 同第 1 の実施例及び第 3 の実施例で適用されるデバッグコマンドの例を示す図。

【図 5】 本発明の第 2 の実施例に係る計算機の機能構成を示すブロック図。

【図 6】 同第 2 の実施例及び第 4 の実施例で適用されるリクエストの原形とデバッグ付きリクエストの例を示す

図。

【図 7】同第 2 の実施例及び第 4 の実施例で適用されるデバッグコマンドの例を示す図。

【図 8】本発明の第 3 の実施例に係る計算機の機能構成を示すブロック図。

【図 9】同第 3 の実施例で適用されるソースプログラム P1 の例を示す図。

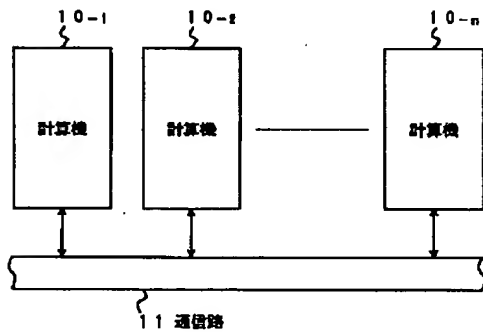
【図 10】本発明の第 4 の実施例に係る計算機の機能構成を示すブロック図。

【符号の説明】

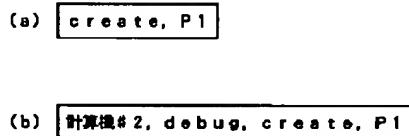
10-1~10-n, 10-i…計算機、11…通信路、20, 30…プログラム位置管理部、22…リクエスト生成部、23, 33, 43, 53…リクエスト通信部、24, 44…デバッグ実行部、25, 35, 45, 55…デバッグ操作部、26, 36, 46, 56…コマンド送

信先選択部、27, 38, 48, 58…デバッグ情報通信部、29, 39, 49, 59…デバッグ情報送信先選択部、31, 51…送信先管理部、32…リクエスト生成部（第 1 の変更手段）、34…デバッグ実行部（第 2 の変更手段）、40, 50…プログラム管理部、42, 52…プログラムコンパイル部、54…デバッグ実行部（第 1 の変更手段、第 2 の変更手段）、231, 331, 431, 531…リクエスト送信部、232, 332, 432, 532…リクエスト受信部、251, 351, 451, 551…デバッグコマンド入力部、252, 352, 452, 552…デバッグ情報表示部、271, 371, 471, 571…コマンド送信部、272, 372, 472, 572…コマンド受信部、281, 381, 481, 581…デバッグ情報受信部、282, 382, 482, 582…デバッグ情報送信部。

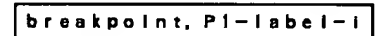
【図 1】



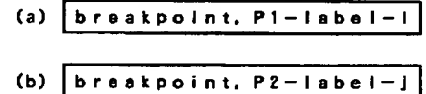
【図 3】



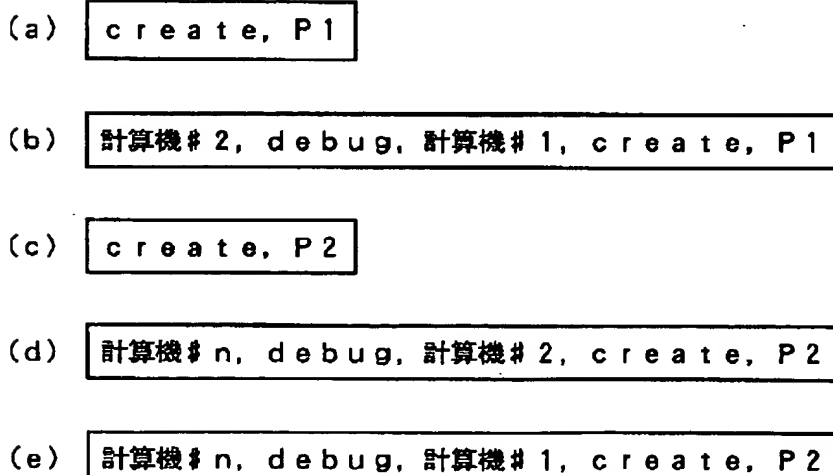
【図 4】



【図 7】



【図 6】



[illegible]

図10-i (i=1~n)

計算機

The diagram illustrates the internal structure of a computer system (計算機) and its communication paths (通信路). The system is organized into several functional blocks and communication channels:

- Internal Components (Left Side):**
  - デバッグコマンド入力部 (Debug Command Input Unit):** 351
  - デバッグ情報表示部 (Debug Information Display Unit):** 352
  - デバッグ操作部 (Debug Operation Unit):** 35
- Internal Components (Right Side):**
  - プログラム位置管理部 (Program Location Management Unit):** 30
  - 送信先管理部 (Destination Management Unit):** 31
  - デバッグ実行部 (Debug Execution Unit):** 34
  - リクエスト生成部 (Request Generation Unit):** 32
  - リクエスト受信部 (Request Reception Unit):** 332
  - リクエスト送信部 (Request Transmission Unit):** 331
  - コマンド受信部 (Command Reception Unit):** 372
  - コマンド送信部 (Command Transmission Unit):** 371
  - デバッグ情報送信部 (Debug Information Transmission Unit):** 382
  - デバッグ情報受信部 (Debug Information Reception Unit):** 381
- Communication Paths (Bottom):**
  - コマンド通信部 (Command Communication Unit):** 37 (encompassing 371 and 372)
  - リクエスト通信部 (Request Communication Unit):** 33 (encompassing 331 and 332)
  - デバッグ情報通信部 (Debug Information Communication Unit):** 38 (encompassing 381 and 382)
- External Communication (Bottom):**
  - 通信路 (Communication Path):** 11 (encompassing the communication units 37, 33, and 38)
- Interconnections:**
  - 351 and 352 are connected to 35.
  - 35 is connected to 31 and 34.
  - 30 is connected to 31 and 32.
  - 31 is connected to 34 and 32.
  - 34 is connected to 32 and 332.
  - 32 is connected to 331 and 382.
  - 332 is connected to 331.
  - 331 and 332 are connected to 33.
  - 372 and 371 are connected to 37.
  - 382 and 381 are connected to 38.
  - 37, 33, and 38 are connected to the external 通信路 (11).

計算機

10-i (i=1~n)

The diagram illustrates a computer system architecture for debugging and communication. It is divided into three main functional areas: Debugging Operation, Program Management, and Communication.

- デバッグ操作部 (Debugging Operation Section):** Located at the top, it contains:
  - デバッグコマンド入力部 (Debug Command Input Unit) 451
  - デバッグ情報表示部 (Debug Information Display Unit) 452
- プログラム管理部 (Program Management Section):** Located in the middle, it contains:
  - プログラム管理部 (Program Management Unit) 40
  - プログラムコンパイル部 (Program Compiler Unit) 42
- デバッグ実行部 (Debug Execution Section):** Located in the center, it contains:
  - デバッグ実行部 (Debug Execution Unit) 44
- 通信部 (Communication Section):** Located at the bottom, it contains:
  - コマンド送信先選択部 (Command Destination Selection Unit) 46
  - コマンド送信部 (Command Transmission Unit) 471
  - コマンド受信部 (Command Reception Unit) 472
  - リクエスト送信部 (Request Transmission Unit) 431
  - リクエスト受信部 (Request Reception Unit) 432
  - デバッグ情報送信先選択部 (Debug Information Destination Selection Unit) 49
  - デバッグ情報送信部 (Debug Information Transmission Unit) 482
  - デバッグ情報受信部 (Debug Information Reception Unit) 481

Connections and Data Flow:

- The **デバッグ操作部** is connected to the **デバッグ実行部** and the **デバッグ情報送信部**.
- The **プログラム管理部** and **プログラムコンパイル部** are connected to the **デバッグ実行部**.
- The **デバッグ実行部** is connected to the **コマンド送信先選択部**, the **リクエスト送信部**, and the **デバッグ情報送信先選択部**.
- The **コマンド送信先選択部** is connected to the **コマンド送信部**.
- The **コマンド送信部** and **コマンド受信部** are connected to the **通信部** via a communication path (47).
- The **リクエスト送信部** and **リクエスト受信部** are connected to the **通信部** via a communication path (43).
- The **デバッグ情報送信先選択部** is connected to the **デバッグ情報送信部**.
- The **デバッグ情報送信部** and **デバッグ情報受信部** are connected to the **通信部** via a communication path (48).

【図9】

```
interface: P1 (Arg1-1, ..., Arg1-N)
interface: P2 (Arg2-1, ..., Arg2-M)
program: P1 (Param1, ..., ParamN)
      :
      fork (P2 (A1, ..., AM) ;
      :
```

【図 10】

